PATENT

SYSTEM AND METHOD FOR NETWORK-BASED AUTOMATION OF ADVICE AND SELECTION OF OBJECTS

Inventors:
Rani Johnson
Scott C. Van Valkenburgh
Anatoly Pekelny

Related Applications

The present application claims priority under 35 U.S.C. §119(e) from provisional application number 60/206122 filed May 22, 2000.

Technical field

The present invention is directed generally to automated advice and selection, and more particularly to a method and apparatus which is able to quickly and efficiently process a large number of rules, based upon a user supplied profile, to provide a categorized set of recommendations and selection criteria, and also to select objects in an efficient and rapid manner from a large inventory of possible objects based upon the categorized set of selection criteria.

Background

Although the present invention will be described in the context of a fashion example, it is to be understood that the concepts and techniques described in this application are applicable to a wide variety of situations in a variety of fields. There is no intent by use of an example in the fashion area to limit the scope of the inventions claimed in this application. It is believed that describing the present invention in the context of a fashion example will render the invention more easily understood, the fashion context being more generally familiar, but nonetheless as complex and variation-intensive as more technologically advanced scenarios.

Dressing oneself is not always easy. Many questions typically run through a person's mind while trying to select clothing. Does this make me look fat? What color shoes go with this outfit? Is this still in style? What should I wear? Tens of thousands of these questions are sent to style columnists across the nation every day, while hundreds of thousands more are asked of retail salespeople. But millions go completely unanswered, resulting in the inquirer choosing apparel that is not right for their body, for their color tones, or for the event they are attending.

Several attempts have been made to connect apparel customers with retailers via the World Wide Web ("Web"). High customer acquisition costs and poor customer retention rates have resulted in disappointing returns for most consumer apparel websites. These poor returns are primarily due to consumers having difficulty in locating precise items, the inherent inability to touch or try on garments, a cumbersome, delivery-based return/exchange process, and the lack of personal assistance. While online apparel shopping offers many unique, interactive possibilities, it can never fully replace visiting a store to shop for clothes.

Most consumer apparel websites are backed by companies that stock and ship apparel directly. These companies do not offer sophisticated, automated advice, nor have robust search capabilities. Many magazines and webzines offer style opinions to their niche audience. However, such advice is neither fully personalized nor comprehensive.

Among the difficulties of offering sophisticated, automated advice is that conventional artificial intelligence methods and systems require sophisticated programming techniques, high performance server systems to process the artificial intelligence applications, and highly trained personnel to administer. The more sophisticated and detailed the user-supplied input, the more complex and computationally intensive the

conventional artificial intelligence solution. Updating or maintaining the domain knowledge for such systems can prove to be arduous tasks.

For example, a conventional method for approaching the fashion advice problem is to use an extensive series of "if-then" statements to address each of the possible combinations of user requirements and clothing attributes. A drawback of such an approach its sheer size and complexity if all feasible combinations are to be handled.

It is therefore desirable to provide an artificial intelligence based-automated advice methodology and system which avoids the cumbersome knowledge representation and heavy computational requirements of conventional approaches, yet can accommodate detailed user requirements and a large number of possible variations in the characteristics of the possible choices. In a fashion context, this artificial intelligence based method and system closely duplicates a clothing and accessory style consultant, also known as a "personal shopper."

Brief Summary of the Invention

The above and other problems and disadvantages of prior automated advice methods and systems are overcome by the present invention of a method, and apparatus therefor, of providing advice and forming criteria based on the advice for selecting objects out of an inventory of available objects. The formulated criteria are based upon user-supplied profile information, a set of object characteristics, and a set of rules which have been formed by associating a set of variations of the object characteristics with a set of variations of input variables. In accordance with the present invention, each variation in object characteristics is associated with each variation in input variables, and a priority is assigned to each such association to form a prioritized rule set. The user-supplied profile information is analyzed to select specific variations from the set of variations of input variables. The selected

input variable variations are applied to the prioritized rule set to obtain a reduced set of prioritized rules. The reduced set of prioritized rules are processed to generate categorized output characteristic values which represent the advice and the criteria for selecting objects.

In a further aspect of the present invention, a method and apparatus are provided for selecting objects from an inventory of objects, each object being described by a set of characteristics and by a value for each characteristic in the set of characteristics, where, for a particular object the assigned values of the characteristics for that particular object are descriptive thereof. In accordance with the present invention, a set of desired characteristic values is formed. A branched path search schema is formed as a function of the desired characteristic values, output characteristic passing criteria, and supplied search order criteria. Objects from the inventory of available objects are evaluated according to the branched path search schema. The evaluated objects are then ranked according to how well the object traversed the branched path search schema.

The present invention provides a straight forward yet sophisticated methodology and structure for accommodating detailed user requirements and a large number of possible variations in the characteristics of the possible choices to provide a set of well-informed recommendations, while avoiding the heavy computational requirements of conventional approaches.

These and other advantages of the present invention will be more readily understood upon considering the following detailed description of the present invention, and the accompanying drawings.

Brief Description of the Drawings

Figure1 is a simplified functional block diagram of the advice engine and the object selection methodology of the present invention

Figure 2 provides a more detailed functional block diagram of the advice engine processing in accordance with the present invention.

Figure 3A and 3B provide a example of the kinds of user profile input data which might be provided in connection with the present invention, and a specific example in the fashion context.

Figures 4A to 4L provide examples of input variables, variations of such variables, and values assigned to such variations of input variables.

Figure 5 illustrates the conversion process by which the user profile input data is used to select particular input variable values.

Figure 6 is an example of a pre-ordered input variable array which is the result of the conversion process illustrated in Figure 5.

Figure 7 illustrates a theoretical knowledge matrix and the relationship between input variables, input variable variations, object characteristics and variations of object characteristics, and assigned priorities.

Figure 8 illustrates the use of the pre-ordered input variable array of Figure 6 to trigger portions of the matrix which are related to the input variable variations set forth in Figure 6.

Figure 9 is an illustration of a theoretical reduced matrix in accordance with the present invention, and demonstrates the relationship between input variables, the object characteristics, and the associated priorities.

Figures 10A to 10Q illustrate object characteristics and variations of such characteristics in the fashion context in the form of pre-defined user input and garment characteristic categories.

Figures 11A to 11D illustrate a reduced matrix and output characteristics for the problem of fashion, and the processing of input variable weights, assigned priorities, and exclusions rules in accordance with the present invention.

Figure 12 provides a more detailed functional block diagram of the object selection methodology in accordance with the present invention.

Figures 13A to 13E illustrate the use of output characteristic values, search order, value and passing standards in accordance with the present invention for the problem of fashion.

Figures 14A to 14D illustrate a branched search engine generated for the output characteristic searching order, values and passing standards set forth in Figures 13A to 13E.

Figures 15A and 15B provide an example of a characterized inventory database for the problem of fashion.

Detailed Description of the Invention

In the specific fashion example described, a system and method are described for apparel advice automation over a network, such as the World Wide Web ("Web").

User of websites typically browse through websites by "clicking" with a computer mouse through a series of strategically organized hyperlinks. On the other hand, consumers of retail outlets browse through apparel by physical walking through "Brick & Mortar" stores that stock the apparel. A system that provides a connection between website users and physical retails stores is preferably referred to as a "Click & Mortar" model. The apparel advice automation described herein will provide the apparel industry with an improved Click & Mortar model for apparel, an extremely "high-touch" product.

At least five primary tools are described to increase apparel websites' "stickiness" and personalization, facilitate specific product searches, drive traffic into Brick & Mortar stores, and create a centralized place for consumers to search for clothing at local outlets. These tools preferably include:

1. An "expert" that supplies highly personalized, occasion specific clothing advice, equal to or better than that of a professional style consultant, which then allows for the purchase of specific clothing choices based on the advice;

2. An industry standard XML ontology and centralized database of detailed product features allowing for extremely specific product searches;

3. Customizable consumer portal software and email notifications that are regularly updated with new inventory, style and seasonal recommendations;

4. A turnkey solution that allows consumers to place an item on hold at a local store to be tried on before purchase, or (depending on the retailer's needs) purchase a garment online then pick it up at a local store;

5. A "portal" based on the aforementioned technologies, the portal allowing consumers to search through a database of products rather than individual stores. This portal can include sticky features such as gifting advice, daily outfit assessments, garment design & find, continually updated information on fashion trends, feedback to designers on their latest lines, discussion groups, chat rooms, expert style columnists, style testimonials, fashion police citations, user's style photo gallery, streaming video of runway shows, and more.

The "expert" identified above will be the primary focus of the detailed description provided herein.

In the fashion example, a "Website" is provided which is centered around the "expert" advice method and apparatus, and is preferably configured to produce comprehensive written reports with illustrations of recommended attire. Clothing experts provide expert information to a database associated with the "Website." These clothing experts work directly with designers to display actual examples of clothing articles in the advice reports. As inventory is added, an extensive database of

well-described products is developed, allowing for precise searches of specific products.

In one version, links are provided to designers' website. Major fashion magazines are engaged by offering free advertising on the Website in exchange for positive articles about the Website. As the Website brand, traffic and credibility builds, retailers may be approached to fulfill the demand generated for the products displayed. Items are delivered through a retailer's existing shipping infrastructure, or a fax is sent to a local store's customer service department to inquire about availability.

To complete the overall solution, appropriate database technologies are utilized for robust integration of local retail inventories with the Website. Ultimately, an application service provider ("ASP") sells the complete service and/or individual technologies to apparel e-tailers, portals, and style webzines. Once registered with a personalized profile, consumers will find their profile on all sites using technology of the present invention.

The present invention may be used in connection with marketing efforts to target people discontent with their physical appearance or with their social/romantic status. The technology may also be used to target online body-conscious women, and single men. Combined, these two groups represent 31 million people.

There are over 100 large apparel retailers in the U.S. along with thousands of smaller stores suitable for using technology of the present invention. Mid-range to high-end department stores, such are also suitable users.

Conceptually, in accordance with the present invention a user is prompted to complete a profile, which the system understands and uses to trigger applicable rules in a knowledge matrix. The triggered rules are summarized to exclude conflicts and determine the output characteristic values (which define the optimal characteristics). In conjunction with the

preset categorized, output characteristic searching order and output characteristic passing standards, these output characteristic values are fed into the searching schema, generating in an individualized search engine for each distinct profile. This search engine queries the characterized inventory database ultimately resulting in prioritized inventory selections (again unique to each profile).

Referring to Figure 1, the present invention will now be described in greater detail. The present invention has two distinct parts which can function independently of one another: an advice engine 10, and an object selection methodology 12.

Advice engine 10 takes in a user input profile 14, uses the information from the user input profile 14 to select input variables 16 which trigger rules in a knowledge matrix 18. In turn, these triggered rules 20 are evaluated and processed in a processing block 22. The result of the processing in block 22 is a set of categorized output characteristic values 24.

The object selection methodology 12 uses information such as the set of categorized output characteristic values 24, a search order 26, and passing criteria 28 in a search schema forming operation 30. The result of the search schema forming operation 30 is a branched path search engine 32 which can be individualized or customized to a particular user or set of circumstances.

Characterizations of objects, such as fashion items which have been characterized and stored in an inventory database 34, are subjected to the branched path search engine 32, evaluated, and ranked. The result is a prioritized inventory selection list 36, which is the output of the object selection methodology and system 12.

Advice Engine -- Criteria Formation

Additional details about advice engine 10 are provided in Figure 2. The user profile input 14 can be an array of information upi(i) as in Figures 3A and 3B, which will be described in detail below. The user

profile input 14 is converted in a conversion process 38 into the select input variables 16 which are formed into a pre-ordered input variable array 40.

In order to form the pre-ordered input variable array 40, the conversion process 38 uses a set of input variables each of which has a number of defined variations. Depending upon information supplied in the user profile input 14, different variations of the input variables will be identified.

The pre-ordered input variable array 40 is applied to knowledge matrix 18 to trigger corresponding portions of the matrix. Knowledge Matrix 18 associates the possible variations of the input variables with the possible variations of the characteristics, and assigns priorities to each combination of input variable variation and characteristic variation.

These triggered portions or rules 20 of knowledge matrix 18 are used to form a "reduced knowledge matrix" 42. The "reduced knowledge matrix" 42 is then evaluated (see function 22, Figure 2) to generate the "categorized output characteristic values" 24.

Figure 3A illustrates an example of an array of user profile inputs, with eighteen (18) elements or pieces of information making up the array. It is to be understood that the number of elements in the array will be determined by the requirements of the particular application and the level of detail desired for the particular advice task.

Figure 3B provides an example of the user profile input array for the fashion example. As can be seen from this example, the information supplied by the user is of the type which will aid in the selection of the objects of interest, in this case garments and fashion accessories. For example, the nature of the specific event, whether, formal, informal, or other, will impact the kinds of garments which would be appropriate. The time of day, as well as the date of the event, will also dictate whether a light weight or heavier material is most suitable. Information about the user's body, both objective and

subjective are, also requested. In other applications, such as advice on consumer electronics selection, or other retail scenarios, the information to be supplied by the user will be different. For example, for the consumer electronics scenario, for audio reproduction equipment, the user will be asked about listening preferences, room sizes, music sources, and the like.

Figure 4A to 4L illustrate possible input variables for the fashion example, and the possible variations which have been defined for each such variable. For example, Figure 4E corresponds to the input variable of "time" and defines three variations: m1 – morning; m2 – afternoon; and m3 – evening. Figure 4K defines the variable age, "age#," and defines eight (8) variations. Some input variables, such as height/weight, "htwt," represent combined or related profile information, while others, such as body type, "btyp," include a subjective element.

Figures 5 and 6 illustrate how the user profile information obtained in Figures 3A and 3B are subjected to several calculations that convert it into pre-defined categories, Figures 4A to 4L, which are in turn assembled into a pre-ordered input variable array, u(j), Figure 6. In the fashion example, illustrated in Figure 6, the pre-ordered input variable array has thirteen elements.

In Figure 5, the user profile input is provided in the left most column. The center column illustrates the calculations. The right-most column illustrates the calculated "input variable" variation. It can be seen, for example, that input variable u[5] has been set equal to "t4." From Figure 4F it can be seen that "t4" is one of the variations of the body type, "btyp," input variable. In Figure 4F, "t4" corresponds to the "well proportioned" variation. Referring back to Figure 5, it can be seen that the "well proportioned" calculation was made using the user profile input of "bust" and "waist" and "hips." Other calculations and the user profile input used for such calculations are shown in Figure 5.

The pre-ordered input variable array of Figure 6 is used to trigger applicable rules in the knowledge matrix 18, see Figure 1. More particularly, the input variable array triggers analogous columns in the knowledge matrix 18, an extensive, weighted, 2 dimensional knowledge matrix that supports all feasible input conditions. In use, this knowledge matrix is populated with real numbers that represent prioritized rules(pr_{ij}), used in calculating the output characteristic value (oc_i) for the expert system. Each column in the knowledge matrix cab be weighted by a variable multiplier (w_i).

Referring to Figure 7, a simplified, conceptual illustration of the knowledge matrix 18 is provided. It is to be noted that the knowledge matrix 18 is arranged in groups of columns and groups of rows. Each group of columns represents an input variable, and the variations for that input variable. Each group of rows represents a characteristic and the variations for that characteristic. At the intersection of each column and row is a "priority." The priority is assigned to indicate the importance of that combination of the particular input variable variation and characteristic variation, with respect to other variations of that characteristic.

For example, in Figure 7, the first group of columns represents an input variable x1, and variations of v1 through v6 of input variable x1. The first group of rows represents characteristic c1, and variations a0 to a3 of characteristic c1. The priority assigned to the combination of x1v1 and c1a0 is a low "p9." On the other hand, the priority assigned to the combination of x1v1 and c1a1 is a relatively high priority of "p2." In this manner, a large number of combinations of input variable variations and characteristic variations are represented in the knowledge matrix 18, and a priority is assigned to each such combination.

Figure 8 illustrates the knowledge matrix 18 of the present invention applied to the fashion example, and the manner in which

triggers from the pre-ordered input variable array 40 of Figure 6 are used to select certain columns from the knowledge matrix 18 for further processing. It is to be noted that the embodiment of the knowledge matrix 18 shown Figure 8 also includes a row which assigns "weights" to each of the input variable variations. As will be described in greater detail herein below, these "weights" can be changed which in turn will affect selection outcome.

Three of the triggers, or input variables, from Figure 6, e1, s1, and m3, are shown in Figure 8. These "trigger" respective columns in the knowledge matrix 18. These and the other "triggered" columns are used to form the "reduced knowledge matrix" 42. See Figure 2. In other words, The triggered columns in the knowledge matrix form a reduced matrix that is likewise affected by variable multiplier. The applicable, non-excluded, prioritized rule values in the reduced matrix are averaged to generate the final output characteristic values. These values dictate which output characteristic is most favorable.

The following equation characterizes the relationship between the knowledge base matrix, input variable and output characteristic array:

$$\mathbf{oc}_i = \frac{\sum_{i=1}^{N}\left(\mathbf{pr}_{ij} \in \mathbf{S}\right) * \mathbf{w}_j}{\sum_{i=1}^{N}\mathbf{pr}_{ij} \forall \left(\mathbf{pr}_{ij} \notin \mathbf{S}\right)}$$

u_j = (triggered) system input variable array

x_i = (comprehensive) system input variable array

pr_{ij} = priority rule values for the knowledge (and reduced) matrix

w_i = weighted multiplier

S $\in$ **R** [1.0...3.0] = predefined range of real numbers that dictate priority in the knowledge matrix 18. Note that for the purposes of the fashion example, the range of real numbers from 1.0 – 3.0 dictate an applicable, non excluding priority value. The real number 0.0 denotes a 'don't care' or 'no effect' priority. The real number 9.0 indicates 'exclude this characteristic entirely.'

$\mathbf{oc}_i$ represents the sum of all triggered prioritized rules pr_{ij} in the row (i), multiplied by the weights w_j of each triggered column u_j . The result of which is divided the number of triggered rows in the set **S** (that contain applicable rule values **R** [1.0...3.0])

Turning to Figure 9, a "reduced knowledge matrix" 42 is illustrated conceptually. Note that there are fourteen (14) columns, thirteen (13) of which correspond to the input variables from the pre-ordered input variable array 40. While the number of columns in reduced knowledge matrix 42 are reduced in comparison to knowledge matrix 18, it is to be noted that the full compliment of characteristic variations (rows) have been preserved.

Figures 10A to 10Q illustrate for the fashion example, the characteristics of the garments of interest, and their variations, which are used to populate the rows of the knowledge matrix 18. For example, Figure 10B represents the "fit" for a garment "top," and uses the symbol "ft." Possible variations of the "garment fit top" characteristic include "ft0 = loose and ftt2 = fitted.

Figure 10J specifies the "garment material" characteristic, and identifies variations such as "mata" = silk; "mat4" = wool; and "mat9" = rayon. Similarly, Figure 10K corresponds to the "garment pattern" characteristic, and has pattern variations including "pat0" = solid; "pat5" = paisley; and "pat8" = other.

Figures 11A to 11D illustrate a reduced knowledge matrix 42 which contains working numbers for the fashion example. Also

illustrated in Figures 11A to 11D is the processing which is performed using the listed priorities and the column weights to obtain output characteristic values 24.

Taking the "nck1" row as an example, it can be seen that the processing includes multiplying the weight for a column by the priority assigned to the row/column combination, and then repeating the operation for all columns, summing the products, and then dividing the sum by the number of non-zero products. In the case of the "nck1" row, there are two non-zero products which result in a 5.5 value for the "nck1" characteristic. From Figure 10C it can be seen that the "nck1" characteristic variation corresponds to a "neck lined" garment characteristic.

In a similar manner, for the "slv6" row the value for the "slv6" characteristic is determined to be "3." From Figure 10G it can be seen that the "slv6" characteristic variation corresponds to a "long sleeve" garment feature.

It is to be noted that when the value of "9" appears as a priority for any of the characteristics, that characteristic is excluded from the output characteristics. Thus, in Figures 11A to 11D, it can be seen that a number of the characteristics are excluded because a "9" appears in at least one of the columns, and such exclusion in indicated by an "excluded" symbol, Ø.

The right-most column in Figures 11A to 11D represents the categorized output characteristic values 24 for the fashion example, which is a result produced by the advice engine in accordance with the present invention. In particular, for the fashion example, this result provides a list of garment characteristics, possible variations for each garment characteristic, and a prioritization for such features and variations. The resulting output characteristics are arranged into predefined categories. The output characteristic in each category with the lowest overall value is defined as optimal. Successively, the

remaining non-excluded output characteristics are prioritized accordingly.

Therefore, for the user whose user profile was provided for the fashion example of Figures 11A to 11D, the garment fit should be "fft2" or normal with a fairly low priority of 8.6; the highest priority variation for garment neck is "nck4," or low-cut with a priority of 3.5; the garment leg should be "leg1" or "bell" with a priority of 2; and so on. See Figures 10A to 10Q.

It is to be noted that a number of different weights have been applied to the columns in the fashion example of Figures 11A to 11D. In this example, the lowest weights represent input variables which are to have the highest impact on the outcome. For example, input variables m3, d2, and b1 have been assigned weights of "1." From Figures 4A to 4L it can be seen that these input variables correspond to: m3 = time of day -- evening; d2 = endowment – average; and b1 = best body feature – arms. Conversely, de-emphasizing weights of "5" were assigned to input variables "h7" and "t4," which represent: h7 = height/weight – tall and thin; and t4 = body type – well proportioned.

Object Selection Methodology

Referring to Figure 12, the objection selection methodology of the present invention will now be described in greater detail. The searching schema utilized in this system is an ordered search. Its organization is dictated by the categorized output characteristic search order 26. This order can be either preset or determined by utilizing the user profile that accesses an additional knowledge base. The output characteristic passing standard 28 sets the maximum output characteristic value permissible for progression to the next category (as dictated by the categorized output characteristic search order 24) in the search schema.

Once an individualized search engine 32 is fashioned from the above information, objects or items from the characterized inventory database are subjected to the individualized search engine 32. As an

object progresses through the individualized search engine 32, a score is kept of how well the item satisfies the search criteria. For example, the score might be incremented for each level successfully passed, and decrement by a like amount for each level not successfully passed.

Figures 13A to 13E provide an example using the problem of fashion for each of search order, passing criteria, and categorized output characteristic values which are used to form the individualized search engine. In the figures, the left-most column identifies the output characteristic category, the second column represents a designated search order for each of the characteristic categories, the third column represents the "output characteristic values" from the advice engine, and the fourth column represents provided "passing standards." For example, the "garment occasion" category is the third priority to be considered in the search. The passing standard for the "garment occasion" category is "4," which rules out garments which are for "occ3," "occ5," and "occ6."

Similarly, for the "garment color tone" characteristic category, the search priority is an "8," indicating that it will be the eight characteristic considered. The passing standard is "5," which result in "tne1" = light, and "tne2" = bold being excluded.

Figures 14A to 14D illustrates the individualized branched path search which was formed from the information in Figures 13A to 13E. Consistent with Figures 13A to 13E, the "garment gender" characteristic category 44 is searched first, followed by the "garment type" category 46. Thereafter, "garment occasion" 48 and then "garment season" 50 are searched, all in accordance with the "search order" column in Figures 13A to 13E.

In Figures 14A to 14D, the bolded characteristic variations indicate ones which meet the "passing standard" for that characteristic. Thus, for the "garment occasion" block, only "occ1" and "occ2" are bolded in view of the indicated passing standard of "3." These bolded

characteristics indicated the possible valid paths that can be taken through the search level. The non-bolded characteristics are considered to be excluded from the possible paths which may be taken through the search level.

This individualized search engine 32 of Figures 14A to 14D queries the characterized inventory database 34, accumulating the output characteristic values for its corresponding path. The characterized inventory that does not map directly to the path dictated by the search engine accumulates a penalty for every non-matching stage. The result of the search engine's query is a score for each inventory item that represents how well it maps to the optimal output characteristics.

Figure 15A and 15B illustrate a characterized inventory database which may be queried by the search engine 32 of Figures 14A to 14D. (In these figures, the number "0" represents a "don't care" or "no effect" priority, and the number "9" represents an "exclude this characteristic entirely" indication.) For example, examining the second item in the inventory, starting from the "garment type" characteristic 46, it can be seen in Figure 15A that all of the garments in the inventory are type 1 and type3, which satisfies the "garment type" characteristic 46. For the next characteristic to be checked, "garment occasion," the second item in Figure 15A is a type 2 or type 4, which meets the criteria. In this manner, the garments in inventory are queried by the search engine 32, and a prioritized inventory selection 36 is provided.

Because of the efficient structure of the advice engine 10 and the search engine 12 of the present invention, an advice system and object locating methodology is provided which is quick and flexible. The system of the present invention is also scalable, and can support the addition of numerous rules on an ongoing basis as the system is improved to provide increasingly more detailed advice. Further, because of its simplicity, the present invention can support to addition or changes in

input and output variable (for example, as additional garment and accessory items are added).

As can be appreciated from the foregoing description of the present invention, customization of rules for individual user or e-tailer's needs (*i.e., an e-tailer may want to increase the likelihood that a certain garment is recommended*), as well as an ability to add and change different rules as seasons and trends change, can be readily accommodated. Changes in fashion trends can be reflected in the priorities given to each characteristic/input variable combination; and weights given to the input variables can be used make further refinements as fashion trends shift the emphasis to different features. Changes in search order as well as the passing criteria can also be used to alter the advice given by advice engine 10, and the garments selected by selection methodology 12.

It can also be appreciated that because of the architecture of the present invention, additions and deletions from the inventory database are simple and easy to make.

The present invention is particularly suitable to be implemented in a conventional personal computer, web server, or the like.

As can be appreciated from the foregoing, the system and method of the present invention, as illustrated in the network based automation of apparel advice and selection embodiment, is fast, efficient, expandable, scaleable, maintainable, reusable and suitable for solving a wide variety of other complex, real world problems.

It is to be understood that the method and apparatus of the present invention, while described in the context of a retail fashion example, is equally applicable and suitable for use in a wide variety of other areas. For example, the present invention can be used in specifying and selecting components in the electronics industry based upon user-supplied required features, performance and cost. Other applications or uses of the present invention include the other retail

scenarios, or any situation where many variables and variations must be applied to many possible choices, in the context of a large body of selection rules. The present invention is likewise capable of incorporating feedback loops to support iterative or real time thinking scenarios.

Attached hereto on pages 51 through 71 is an Appendix of code listings, data and definitions, which provide further detail about the fashion example of the present invention.

It is to be understood that the term "objects" as used herein can refer to anything that has characteristics associated with it. An example might be an army moving across a battlefield and a characteristic might be it's speed, direction, size, etc. Therefore, the term "object" is not meant to be limited solely to physical or inventory objects. The system could be used to just create best parameters for an "object" at any given time.

The present invention has been described above with reference to a fashion embodiment. However, those skilled in the art will recognize that changes and modifications may be made in the above described embodiments without departing from the scope of the invention. For example, the present invention is applicable to any scenario in which a large number of decisional rules, characteristics, and input variables are involved. Furthermore, while the present invention has been described in connection with a specific processing flow, those skilled in the are will recognize that a large amount of variation in configuring the processing tasks and in sequencing the processing tasks may be directed to accomplishing substantially the same functions as are described herein. These and other changes and modifications which are obvious to those skilled in the art in view of what has been described herein are intended to be included within the scope of the present invention.

APPENDIX

CONTENTS

1) screenshot of the USER INPUT PROFILE (figure 1- 14),

2) the code that converts the user input profile to usable info for the knowledge matrix (figure 1-16)

3) the KNOWLEDGE MATRIX (figure1- 18)

4) the code that does the PROCESSING FOR CATEGORIZED OUTPUT -CHARACTERISTIC VALUES (figure1- 22)

5) the CATEGORIZED OUTPUT -CHARACTERISTIC SEARCH ORDER (figure1- 26)

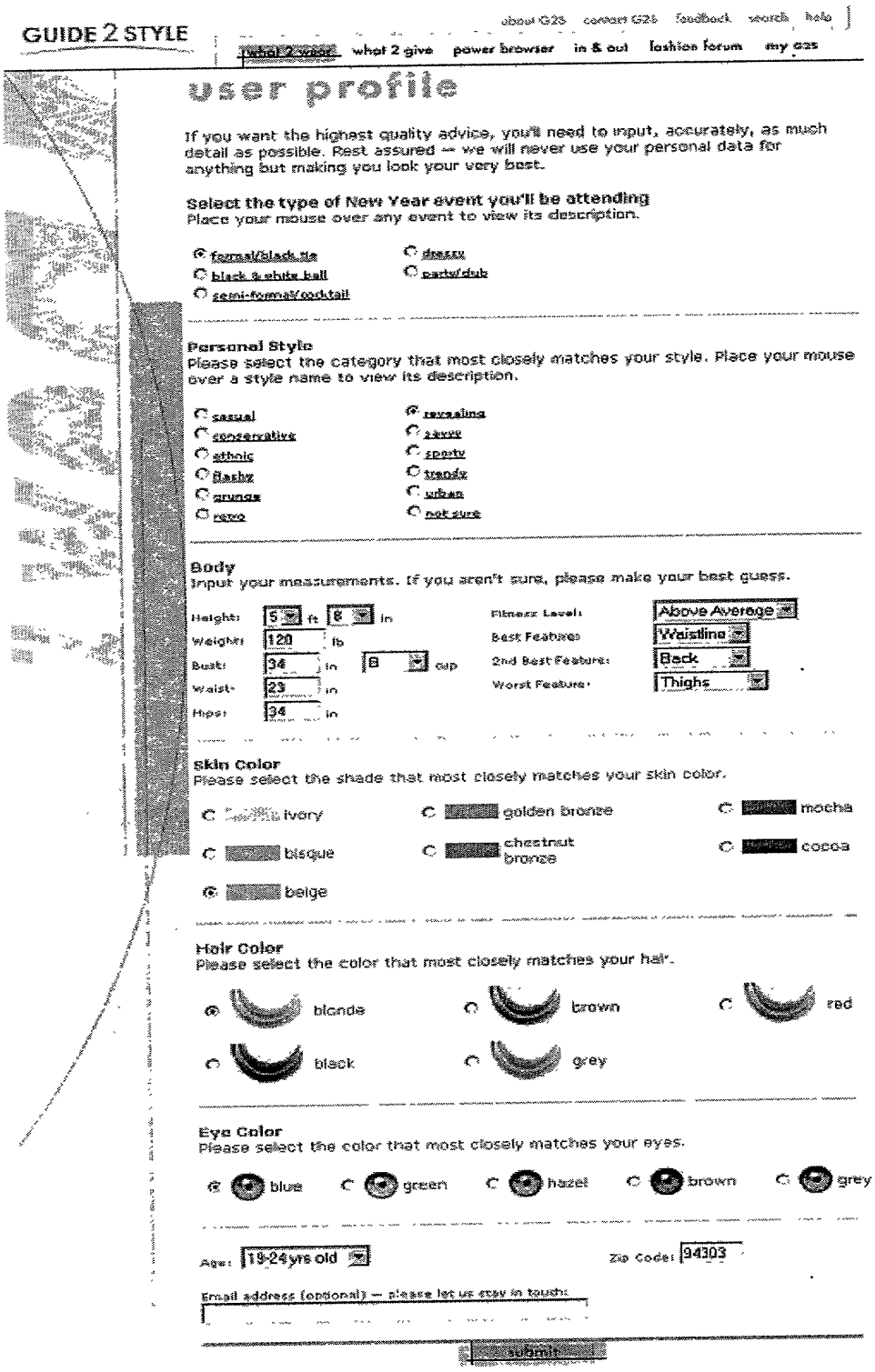
6) the OUTPUT CHARACTERISTIC PASSING CRITERIA (figure1- 26)

7) the SEARCHING SCHEMA/ RULES (figure1- 30)

8) the CHARACTERIZED INVENTORY DATABASE(figure1- 34)

9) screenshot of the advice and PRIORITIZED INVENTORY SELECTIONS (figure1- 36)

1) screenshots of the USER INPUT PROFILE (figure 1- 14):



2) the code that converts the user input profile to usable information for the knowledge matrix (figure 1-16):

```
/**
 *  Clean Version 3.0
 *  Works with GuideAll Version 3.0
 *  and GetScore3 Version 3.0
 *  Keep Vectors for both Garmets files and
 *  full sorted scores for Garments.
 *  Accept user parameters through constractor.
 *
 */
// package guide;

// import util.Arguments;
import SortAttribs;
import MyComparator;
import java.util.*;
import java.io.*;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;



public class GetScore2 {


        private static final int maxCol = 83;
        private static final int maxRow = 21;
    private static final int maxChoice = 12;
    private static final int maxSheCol = 176;

    public static final int maxAttrib = 21;

        public int[] userChoiceI = new int[maxChoice+1];

        private Vector[] tokenVector = new Vector[maxRow+1];
    private Vector[] choiceVector = new Vector[maxRow+1];
        public Vector[] resultVector = new Vector[maxAttrib];
        public int[] garmentNumber = new int[20];
    private Vector attribFullName = null;
        public int garmentReturns = 0;
        private int maxRet = 21;
        private int maxGType = 7;

    public  Vector[] recGarmentType = new Vector[10];
        public LineNumberReader lnReader = null;
    private String[] attribName = new String[] {"gt1gt0","gt1gt2","gt1gt3","gt1gt4",

        "gt1gt5","gt6gt1","gt2gt0","gt2gt3",

        "gt2gt4","gt2gt5","gt6gt2","gt3gt0",

        "gt3gt4","gt3gt5","gt6gt3","gt4gt0",
```

```
        "gt4gt5","gt6gt4","gt5gt0","gt6gt5",

        "gt6gt0"};
        private String[] userInputShortName = new String[] {"evnt"/* 0 */,"sesn"/* 1
*/,"time"/* 2 */,
                                                "htwt"/* 3 */,"btyp"/* 4 */,"endw"/* 5 */,

                        "bfit"/* 6 */,"bst1"/* 7 */,"bst2"/* 8 */,

                        "wrst"/* 9 */,"sty1"/* 10 */,"age"/* 11 */,

                        "she"/* 12 */};
        private String[] userVarsName =  new String[30];

//      {"evnt"/* 0 */,"sesn"/* 1 */,"time"/* 2 */,"feet"/* 3 */,
//                              "inchs"/* 4 */,"pounds"/* 5 */,"waist"/* 6 */,
//
        "bust"/* 7 */,"cup"/* 8 */,"hips"/* 9 */,"bfit"/* 10 */,
//
        "bst1"/* 11 */,"bst2"/* 12 */,"wrst"/* 13 */,
//
        "styl"/* 14 */,"age"/* 15 */,"hair"/* 16 */,
//
        "skin"/* 17 */,"eye"/* 18 */,"zipcode"/* 19 */};

    public String[]  userProfile = new String[maxChoice+1];
    private int[] userInputGroupNumber = new int[] {13, 4, 3, 9, 5, 4, 3, 7, 7, 9, 11, 8 };
        public String[] userVarsInput = new String[19];
        public Vector sheColArray = null;

        public Properties props = null;

        private String switch1 = "no";
        private String switch2 = "no";
        public String switch3 = "allchoices";
        private String switch4 = "no";

        private boolean printYes = false;
        private int rowN = 0;
        private String  propFile;

        public Vector colArray = null;
        private Vector rowArray = null;
        public Logger logger;

    GetScore2 (Properties props,  String[] userArray) {

       this.props = props;
                this.userVarsName = userArray;
        }

     /**
          *  get Column and Row information
          *  create column/Row arrays
          */
```

```
        public void getColumn()
        {
                        colArray = new Vector();
                        for (int i =0; i<maxCol;i++ ){
                           String gString = "guide.col"+i;
                           colArray.add(props.getProperty(gString, null));
                        }
                        // she column
                        sheColArray = new Vector(maxSheCol);
                        for (int i =0; i< maxSheCol;i++ ){
                           String gString = "guides.shycol"+i;
                           sheColArray.add(props.getProperty(gString, null));
                        }
//              String logfile1 = props.getProperty("log1.file","c:/temp/style.log");
//      logger1 = new SimpleLogger(logfile1);
//
//      logger.debug(2,"In getScore2 constructor");
/*
                String logfile = props.getProperty("log.file","c:/temp/style.log");
      logger = new SimpleLogger(logfile);
                logger.setLevelFilter( Logger.DEBUG );
      logger.debug(2,"In getScore2 constructor");
*/
    }

   /**
        *  create buffer reader for read actual data
    */

        public void readData()
        {
                try
                {
//          switch1 = props.getProperty("attrib.bestscore", "no");
                  switch2 = props.getProperty("attrib.allscore", "no");
                  switch3 = props.getProperty("user.from", "allchoices");
                  switch4 = props.getProperty("choice.show", "no");

                  String dir = props.getProperty("garment.directory", "d:/temp");
                  String fl1 = props.getProperty("garmenttype.file", "dresses.txt");
                  File dataList1 = new File(dir,fl1);
                  BufferedReader listIn1 = new BufferedReader(
                     new FileReader(dataList1));
             lnReader = new LineNumberReader(listIn1);

                }
                catch(FileNotFoundException e)
                { // Stream creation exception
              System.err.println(e);
                  return;
             }
                catch(IOException e)          // File read exception
             {
                  print("Error reading input file" + e );
              return;
```

```
                }
        }

        public void findUserColumns()
        {
                if (switch3.equals("allchoices")) {

                        /**
                         * Provide column numbers for all inputs parameters
                         *  criate input numbers array
                         *  all entries  are from lookup tables as position in an appropriate
array
                         */
                         // find derived variables

        // find height/weight group number
        findHeihtWeight();

        // find body type and endowement based on waist,
                        // bust and hips dimensions
        findBodyTypeEndw();

                        // Find skin/hair/eye abbriviation
                        findHSEnumber();


                   // Find rest input numbers
                        findRestInput(userChoiceI, colArray);

                }
                else {
                        String[] uChoice = new String[maxChoice];
                   for (int i=0;i<maxChoice ;i++ ) {
                                String index = "user.choice"+i;
                                uChoice[i] = props.getProperty(index, null);
                             userChoiceI[i] = Integer.parseInt(uChoice[i]);
//          logger.debug(2," User choice "+uChoice[i]+" for i "+i+" Position Number
"+userChoiceI);
                        }
            }
                 // print uChoice
//                String[] uChoice = new String[maxChoice];
                 for (int i=0;i<maxChoice+1 ;i++ ) {
                                String index = "user.choice"+i;
//                              uChoice[i] = props.getProperty(index, null);
//                           userChoiceI[i] = Integer.parseInt(uChoice[i]);
//          logger.debug(2," User choice "+userChoiceI[i]+" for i "+i);
                 }
        }

        /**
         *    Find height/weight group according to user input
    */
     public boolean findHeihtWeight()
          {
              boolean goodHeghtWeight = true;
```

```
        String  feetS = userVarsName[3];
                        String  inchS = userVarsName[4];
              int[] wgPoints = new int[] { 110,112,114,117,120,123,126,129,132,
                                           135,138,141,144,147,150,155,165,501 };
        String[] wgRanges = new String[] {"wt1","wt2","wt3","wt4","wt5","wt6",

"wt7","wt8","wt9","wt10","wt11","wt12",

"wt13","wt14","wt15","wt16","wt17","wt18"};
                        int inchI = Integer.parseInt(inchS.substring(2, inchS.length()));
                        int inI = Integer.parseInt("5");
                        String  poundS = userVarsName[5];
        int poundI = (int)Double.parseDouble(poundS);
                        if (feetS.equals("ft4") && inchI < 10 ){
                            inchS = "in10";
                        }

                        else if (feetS.equals("ft6") && inchI > 0 ){
                            inchS = "in0";
                        }
        // find weight range
                        for (int i=0;i < 18; i++){
          if (poundI < wgPoints[i] ) {
                                    poundS = wgRanges[i];
                                        break;
          }
           }

                        String fip = feetS+inchS+poundS;
                        // find height/weight group
                        String htwt = props.getProperty("user."+fip,"invalid_htwt");
                        userProfile[3] = htwt;
                        if (htwt.equals("invalid_htwt")){
                                goodHeghtWeight = false;
                        }

        return goodHeghtWeight;
          }

          /**
            * Find body type and endowement based on waist,
                * bust and  hips dimensions
                */
    public boolean findBodyTypeEndw()
        {
        boolean goodBodyEndw = true;
              double[] bwPoints = new double[] { 1.275, 1.30, 1.35, 1.40, 1.45, 1.50, 3.0 };
              double[] hwPoints = new double[] { 1.275, 1.30, 1.35, 1.40, 1.45, 1.50, 3.0 };
        String[] bwRanges = new String[] { "bw1",  "bw2",  "bw3",  "bw4",  "bw5",  "bw6",
"bw7" };
        String[] hwRanges = new String[] { "hw1",  "hw2",  "hw3",  "hw4",  "hw5",  "hw6",
"hw7" };
                    int[]   brstPoints = new int[]   { 30, 32, 34, 36, 38, 40, 71 };
                    String[] brstRanges = new String[]
{"br1","br2","br3","br4","br5","br6","br7"};
```

```
String waistS = userVarsName[6];
String bustS = userVarsName[7];
String cupS = userVarsName[8];
String hipS = userVarsName[9];
        String bwRange = null;
        String hwRange = null;
        String brstRange = null;

double waistF = Double.parseDouble(waistS);
double bustF = Double.parseDouble(bustS);
double hipF = Double.parseDouble(hipS);

        double buwaF = bustF/waistF;
        double hiwaF = hipF/waistF;
int i=0;
        for (i=0;i < 7; i++){
   if (buwaF < bwPoints[i] ){
                bwRange = bwRanges[i];
                    break;
   }
   }
        for (i=0;i < 7; i++){
   if (hiwaF < hwPoints[i] ){
                hwRange = hwRanges[i];
                    break;
   }
}
String btyp = props.getProperty("user."+bwRange+hwRange,"invalid_btyp");
userProfile[4] = btyp;
        if (btyp.equals("invalid_btyp")){
                goodBodyEndw = false;
        }

// endowment
int brstI = (int)bustF;
        for (i=0;i < 7; i++){
    if (brstI < brstPoints[i] ){
                brstRange = brstRanges[i];
                    break;
    }
}
        String endw =
props.getProperty("user."+brstRange+cupS,"invalid_endow");
        userProfile[5] = endw;
        if (endw.equals("invalid_endow")){
           goodBodyEndw = false;
        }
return goodBodyEndw;
    }
/**
     * Find hair/skin/eye combination
     */
public boolean findHSEnumber()
     {
       boolean goodShe = true;
       String retSHE = null;
```

```
                String h = userVarsName[16];
                h = h.substring(1,h.length());
                String s = userVarsName[17];
                s = s.substring(1,s.length());
                String e = userVarsName[18];
                e = e.substring(1,e.length());
//              logger.debug(2,"h,s,e: "+h+","+s+","+e);
                int i=0;
                // hair
                String hair = props.getProperty("guides.hair"+h, "radiculess");
                if (hair.equals("radiculess"))
                        goodShe = false;
                String skin = props.getProperty("guides.skin"+s, "radiculess");
                if (skin.equals("radiculess"))
                  goodShe = false;
                String eye = props.getProperty("guides.eye"+e, "radiculess");
                if (eye.equals("radiculess"))
                  goodShe = false;
        if (goodShe)
        {
                retSHE = h+s+e;
                        userProfile[12] = retSHE;
        }
//              logger.debug(2, " hair, skin, eye "+hair+","+skin+","+eye+" FOR SHE
"+retSHE);
                return goodShe;
    }


//          {"evnt"/* 0 */,"sesn"/* 1 */,"time"/* 2 */,
//          "htwt"/* 3 */,"btyp"/* 4 */,"endw"/* 5 */,
//          "fit"/* 6 */,"bst1"/* 7 */,"bst2"/* 8 */,
//          "wrst"/* 9 */,"sty1"/* 10 */,"age"/* 11 */,
//          "she"/* 12 */};

// {"evnt"/* 0 */,"sesn"/* 1 */,"time"/* 2 */,"feet"/* 3 */,
// "inchs"/* 4 */,"pounds"/* 5 */,"waist"/* 6 */,
//  "bust"/* 7 */,"cup"/* 8 */,"hips"/* 9 */,"bfit"/* 10 */,
//  "bst1"/* 11 */,"bst2"/* 12 */,"wrst"/* 13 */,
//          "styl"/* 14 */,"age"/* 15 */,"hair"/* 16 */,
//          "skin"/* 17 */,"eye"/* 18 */,"zipcode"/* 19 */};

            /**
              *  Find rest input number from arrays of templates
              */
    public void findRestInput(int[] uChoice, Vector cArray){
                        String choice = "none";
                        int i,j,k;
                        // fild rest of userProfile array
                        userProfile[0] = userVarsName[0];
                        userProfile[1] = userVarsName[1];
                        userProfile[2] = userVarsName[2];
                        userProfile[6] = userVarsName[10];
                        userProfile[7] = userVarsName[11];
                        userProfile[8] = userVarsName[12];
                        userProfile[9] = userVarsName[13];
```

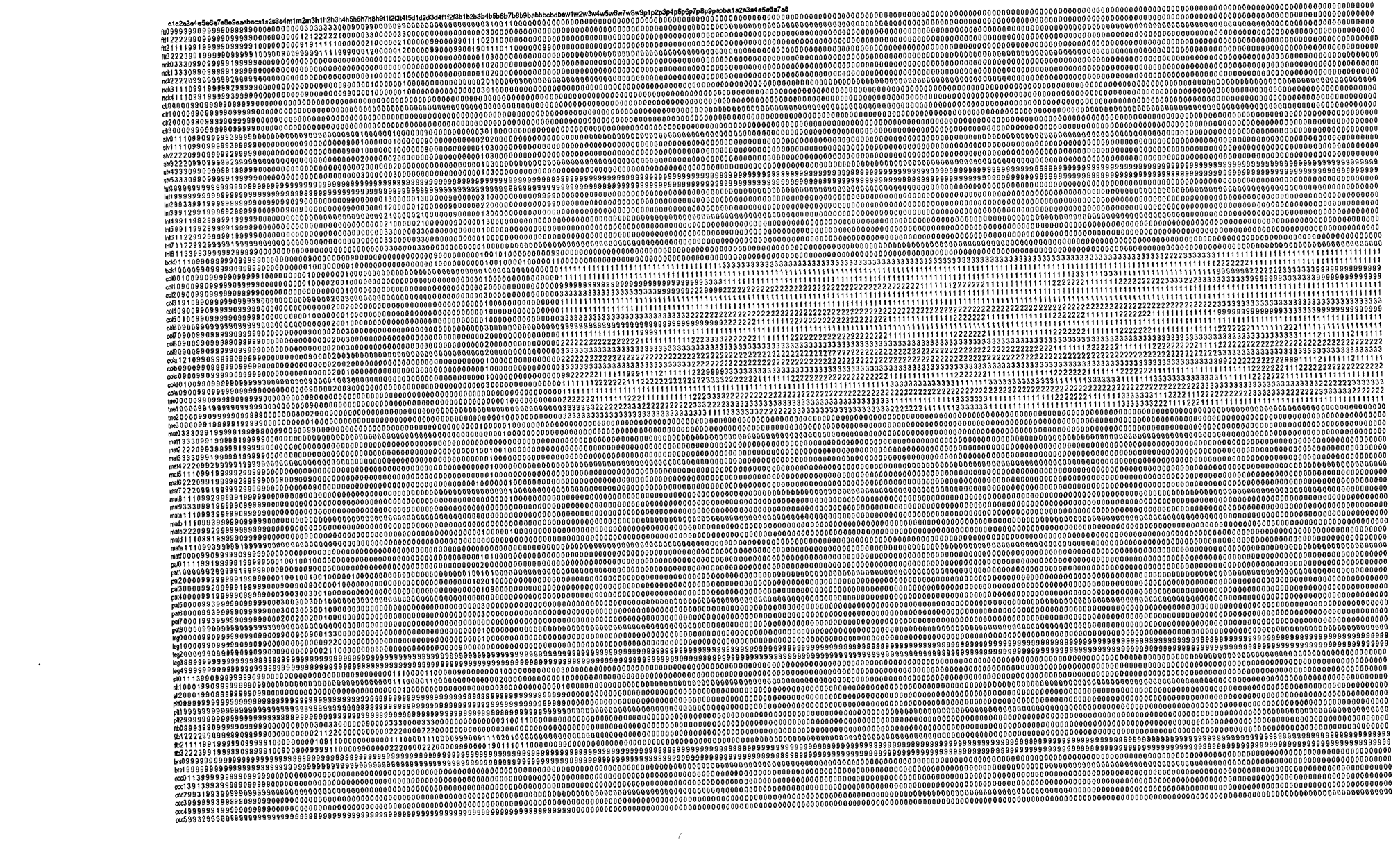
```
                        userProfile[10] = userVarsName[14];
                        userProfile[11] = userVarsName[15];
        int[] colIndex = new int[] {1,13,17,20,29,34,38,41,48,55,64,75};
                        for (i=0; i< maxChoice ;i++ ){
                            choice = userProfile[i];
                                    // find it position in arrays of columns
                                    for (k=colIndex[i];k < cArray.size(); k++ ){
                                        if
(cArray.elementAt(k).toString().equals(choice)){
                                                uChoice[i] = k;
                                                break;
                                        }
                                    }
        }
                        // add SHE index
                        for (j=0;j < maxSheCol ;j++ )
                        {
                            choice = userProfile[12];
                            if (sheColArray.elementAt(j).toString().equals(choice))
                            {
                                        uChoice[12] = j+82;
                                        break;
                                }
                        }
         }
         /**
          *   Main function to read and parse data file
          */
         public void getData()
         {

    String lineText = null;
          int lineCount = 0;
          int tokenCount = 0;
    String[] tArray = new String[100];
          for (int k=0;k<100 ;k++ )
    {
           try
                {
               lineText = lnReader.readLine();
     } catch (IOException e) {
                        System.out.println( "IO error reading data file. " + e.toString() );
                        System.exit(11);
                }
           if (lineText == null)
                {
                   break;
           }
     if (lineText.indexOf("DATA_START") != -1)
     {
                        continue;
     }
     if (lineText.indexOf("DATA_END") != -1)
     {
                        break;
     }
```

```
                lineCount++;
                StringTokenizer st = new StringTokenizer(lineText);
                tokenCount = 0;
                while (st.hasMoreTokens())
                {
                    String token = st.nextToken();
                        tArray[tokenCount] = token;
                        tokenCount++;
    }
    int cntNot0=0;
                float finalScore = (float)0.0;
                cntNot0 = 0;
                boolean isNull = false;
                choiceVector[k] = new Vector();
                choiceVector[k].insertElementAt(tArray[0],0);
//if (lineCount == 15)
//{
//print(" Line "+lineText);
//}
    String sw = null;
    int wf = 1;
                for (int i=0;i<12 ;i++)
    {
                    sw = props.getProperty("applied.to.garment","0")       ;
                        if(sw.equals("1"))
                          wf =
Integer.parseInt(props.getProperty("weighted.attrib."+i,"1"));
                    String score = tArray[userChoice![i]];
                        print("wF "+wf+" i "+i);
                        if (score.equals("9"))
                        {
                     isNull = true;
                          choiceVector[k].insertElementAt(new Float(9.0),1);
                          break;
                        }
      if (!score.equals("0"))
      {
                           cntNot0++;
      }
                        int partScore = Integer.parseInt(score)/wf;
                    finalScore += partScore;

        }
        if (isNull==false) {
          if (cntNot0==0){
           choiceVector[k].insertElementAt(new Float(5.0),1); // Zero count
          }
          else {
                float realScore = (float)finalScore/cntNot0;
            choiceVector[k].insertElementAt(new Float(realScore),1);
          }
        }
      }
  }  // end of read info
```

```
        /**
         *    Print final scores
         */

        public void printFinalScore()
        {
                        for (int i=0;i<maxRow ;i++ )
                {
                        String result = choiceVector[i].elementAt(1).toString();
                        if (result.indexOf(".") != -1)
                        {
                                result = result.substring(0, result.indexOf(".")+2);
                        }

                }
        }
    /**
         *  Sort garments according their Scores
         */
        public void multChoice(int aInd, int aRet)
        {
       int aIndex = 0;
       int aNumb = maxAttrib;
                SortAttribs[] sa = new SortAttribs[aNumb];

            for (int i=0;i<aNumb ;i++ )
                {
                        String fName = attribName[aIndex+i].toString();
                        float attf =
Float.parseFloat(choiceVector[i].elementAt(1).toString());
                        Arrays.fill(sa, i, i+1, new SortAttribs(attf, fName));
                }
                        Arrays.sort(sa,new MyComparator());
                        for (int i=0;i<aNumb ;i++ )
                        {
//                          logger.debug(579,sa[i].toString());
                        }

                        for (int i=0;i<maxRet ;i++)
                        {
                          float fl = sa[i].getX();
                                if (fl < 9.0)
                                {
                                  resultVector[garmentReturns] = new Vector();
                                  String s = ""+fl;
                                  s = s.substring(0, s.indexOf(".")+2);
                                  resultVector[garmentReturns].add(0, sa[i].getName());

//          print("name "+sa[i].getName()+" score "+fl);
                                  resultVector[garmentReturns].add(1,s);
                                  garmentReturns++;
                                }
                        }

      }
    /**
```

```
        * Converted Garments Combination found to the garments type
        * and files
        */
   public void findGarments()
        {
                for (int i=0;i < garmentReturns ;i++ )
                {
                        String garmComb = resultVector[i].elementAt(0).toString();  // like
gt1gt3
                        String garm1 = garmComb.substring(0,3);    // like gt1
                        String garm2 = garmComb.substring(3,garmComb.length());  //
like gt3
                        int gn1 = Integer.parseInt(garm1.substring(2,garm1.length()));
//like 1
                        int gn2 = Integer.parseInt(garm2.substring(2,garm2.length()));
//like 3
                        garmentNumber[2*i] = gn1;
                        garmentNumber[2*i+1] = gn2;
        recGarmentType[i] = new Vector();
                        String gtype =
props.getProperty("garment."+garm1,"nonesence");
                        if (!gtype.equals("nonesence"))
                        {
                            recGarmentType[i].add(gtype);  // like shirts

                        }
                                gtype =
props.getProperty("garment."+garm2,"nonesence"); // like skirts
                                if (!gtype.equals("nonsence"))
                                {
                                    recGarmentType[i].add(gtype);

                                }
                }

        }
        public void print(String s)
        {
     System.out.println(s);
        }


}
```

3) the knowledge matrix (figure1- 18)



```
#weighted inputs
# switches 0 - not applied, 1 applied
applied.to.garment=0
applied.to.gfeatures=1

#user specific events: e1-ec
weighted.attrib.0=3
#season: s1-s4
weighted.attrib.1=5
#time of day: m1-m3
weighted.attrib.2=5
#height/weight ratio: h1-h0
weighted.attrib.3=2
#body type: t1-t5
weighted.attrib.4=1
#endowment: d1-d4
weighted.attrib.5=3
#body fitness level: f1-f3
weighted.attrib.6=3
#best feature 1: b1-b7
weighted.attrib.7=3
#best feature2: b8-be
weighted.attrib.8=4
#worst body feature: w1-w9
weighted.attrib.9=3
#personal style: p1-pb
weighted.attrib.10=2
#user age range: a1-a8
weighted.attrib.11=4
#she: 111-175
weighted.attrib.12=2
```

4) the code that does the PROCESSING FOR CATEGORIZED OUTPUT - CHARACTERISTIC VALUES (figure1- 22):

```
/**
 * Final version 3.0
 * Working and clean.
 * Generates sorted arrays of all scores.
 */
// package guide;
//import SortAttribs;
//import MyComparator;
//import GetGarment;
// import util.Arguments;
import java.util.*;
import java.io.*;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;


public class GetScore3 {


    private static final int maxCol1 = 83;
    private static final int maxCol2 = 176;
    private static final int maxRow = 108;
    private static final int maxChoice1 = 12;
    private static final int maxChoice2 = 1;
    private static final int maxChoice = 13;

    public static final int maxAttrib = 17;

    public static final int upperInd = 0;
          public static final int neckInd = 1;
          public static final int clrInd = 2;
          public static final int sleevInd = 3;
          public static final int lintInd = 4;
          public static final int backInd = 5;
    public static final int colInd = 6;
          public static final int tneInd = 7;
          public static final int matInd = 8;
          public static final int patInd = 9;
          public static final int legInd = 10;
          public static final int slitInd = 11;
          public static final int pleatInd = 12;
          public static final int lowerInd = 13;
          public static final int brasInd = 14;
          public static final int occsnInd = 15;
          public static final int styleInd = 16;

          public static final int upperRet = 1;
          public static final int neckRet = 2;
          public static final int clrRet = 1;
          public static final int sleevRet = 2;
```

```
        public static final int lintRet = 2;
        public static final int backRet = 1;
        public static final int colRet = 6;
        public static final int tneRet = 2;
        public static final int matRet = 4;
        public static final int patRet = 3;
        public static final int legRet = 1;
        public static final int slitRet = 1;
        public static final int pleatRet = 1;
        public static final int lowerRet = 1;
        public static final int brasRet = 1;
        public static final int occsnRet = 1;
        public static final int styleRet = 1;

        private   PrintWriter out1;

        private StringTokenizer st;

    private Vector[] dataRowVector = new Vector[maxRow];

        public Vector[] resultVector1 = new Vector[maxAttrib];
        public Vector[] resultVector2 = new Vector[maxAttrib];

        private Vector[] attribFullName = new Vector[maxAttrib];

        public LineNumberReader lnReader = null;

        private int[] userChoice = new int[maxChoice];

        public int[] attribNumber = new int[] {4,5,4,6,9,2,15,4,16,9,5,3,3,4,2,6,11};
        private int[] attribIndex = new int[maxAttrib];


        private String switch1 = "no";
        private String switch2 = "no";
        private String switch3 = "allchoices";
        private String switch4 = "no";
    private int vectorNumb = 1;
        private int rowN = 0;

        private Vector colArray = null;

    // files passed as a parameters
    protected String dataFile1 = null;

        public int[] garmentNumber = new int[20];

        public Properties props;
        private Logger logger;

    /**
         *  Constructor
         */
    GetScore3 (Properties props, int[] userChoice) {
        this.userChoice = (int[])userChoice.clone();
                this.props = props;
```

```
    }

/**
     * get Column and Row information.
     * create column/Row arrays
     */

    public void getColumn()
    {
                    colArray = new Vector();

                    String gs = props.getProperty("guides.col1", null);
                    String gs1 = props.getProperty("guides.shycol1", null);
                    for (int i =0; i<maxCol1;i++ )
                    {
                       String gString = "guides.col"+i;
                       colArray.add(props.getProperty(gString, null));
                    }
                    for (int i =1; i<maxCol2;i++ )
                    {
                            String gString = "guides.shycol"+i;
                       colArray.add(props.getProperty(gString, null));
                    }
 }


/**
     * create buffer reader for read actual data
 */

    public void readData(String file1)
    {
            try
            {
        switch1 = props.getProperty("attrib.bestscore", "no");
              switch2 = props.getProperty("attrib.allscore", "no");
              switch3 = props.getProperty("user.from", "allchoices");
              switch4 = props.getProperty("choice.show", "no");

  // initialize out printing
  try
            {
                    // creating output files
                    String dir = props.getProperty("out.directory", "d:/temp");
                    String fl  = props.getProperty("out.file1", "Contest1.txt");
                    File f = new File(dir,fl);
               out1 = new PrintWriter(
    new FileWriter(f));
  } catch(IOException e) {}


            String dir = props.getProperty("garment.directory", "d:/temp");
            File dataList1 = new File(dir,file1);
            // Definition for write portion
        String dirName = null;
            BufferedReader listIn1 = new BufferedReader(
```

```
                    new FileReader(dataList1));
            lnReader = new LineNumberReader(listIn1);
                }
                catch(FileNotFoundException e)
                { // Stream creation exception
            System.err.println(e);
                  return;
            }
                catch(IOException e)          // File read exception
            {
                  print("Error reading input file" + e );
             return;
                }
}


/**
        * Get row data from data files
        */

       public void getData()
       {

 String lineText = null;
       int tokenCount = 0;
 String[] tArray = new String[300];

       for (int j=0,k=0;j<150 ;k++,j++ )
 {
        try
                {
            lineText = lnReader.readLine();
  } catch (IOException e) {
                        System.out.println( "IO error reading data file. " + e.toString() );
                        System.exit(11);
                }

        if (lineText == null)
                {
                  break;
        }
  if (lineText.indexOf("DATA_START") != -1)
  {
                  k--;
                        continue;
  }
  if (lineText.indexOf("DATA_END") != -1)
  {
                        break;
  }

                StringTokenizer st = new StringTokenizer(lineText);
                tokenCount = 0;
                while (st.hasMoreTokens())
                {
                    String token = st.nextToken();
```

```
                        tArray[tokenCount] = token;
                        tokenCount++;
    }
    int cntNot0=0;
            float finalScore = (float)0.0;
            cntNot0 = 0;
            boolean isNull = false;
            dataRowVector[k] = new Vector();
            dataRowVector[k].insertElementAt(tArray[0],0);
            String sw = null;
    int wf = 1;
            for (int i=0;i<maxChoice ;i++)
    {
              sw = props.getProperty("applied.to.gfeatures","0")      ;
                  if(sw.equals("1"))
                        wf =
Integer.parseInt(props.getProperty("weighted.attrib."+i,"1"));
                  String score = tArray[userChoice[i]];
                        if (score.equals("9"))
                        {
                   isNull = true;
                         dataRowVector[k].insertElementAt(new Float(9.0),1);
                         break;
                        }
      if (!score.equals("0"))
      {
                          cntNot0++;
      }
                  finalScore += (float)Integer.parseInt(score)/wf;
                        out1.println(" finalscore: "+finalScore+" wf: "+wf+ " score:
"+score+" i: "+i);
                        out1.flush();

                        if (k > 29 && k < 45)
                        {
                           String color = props.getProperty("garment.attrib"+k,"none");
                                String colmn =
props.getProperty("guides.col"+userChoice[i],"none");
//                          logger.debug(3," score for color = "+color+" equals
"+finalScore+" under column "+colmn);
                        }
          }

          if (isNull==false) {
            if (cntNot0==0){
             dataRowVector[k].insertElementAt(new Float(5.0),1); // Zero count
            }
            else {
                  float realScore = (float)finalScore/cntNot0;
               dataRowVector[k].insertElementAt(new Float(realScore),1);
            }
          }
        }
  }   // end of read info
```

```
/**
        *  Get the best scores
        */

        public void getMultChoice(int vectorNumb)
        {
    this.vectorNumb = vectorNumb;
                attribIndex[0] = 0;

          for (int i=0;i<maxAttrib ;i++ )
          {
                  if (i<maxAttrib-1)
                      {
                              attribIndex[i+1] = attribIndex[i]+attribNumber[i];
              }
                }

          for (int i=0;i<maxAttrib ;i++ )
          {
                  attribFullName[i] = new Vector();
                      for (int j=0;j<attribNumber[i];j++ )
                      {

attribFullName[i].insertElementAt(props.getProperty(""+i+".garment.attrib"+j,null),j);
//                              logger.debug(3, " i: "+i+", j: "+j+" FullName
"+props.getProperty(""+i+".garment.attrib"+j,null));
                      }
          }

                      multChoice(upperInd, upperRet);
                      multChoice(neckInd, neckRet);
                      multChoice(clrInd, clrRet);
                      multChoice(sleevInd, sleevRet);
                      multChoice(lintInd, lintRet);
                      multChoice(backInd, backRet);
                      multChoice(colInd, colRet);
                  multChoice(matInd, matRet);
            multChoice(tneInd, tneRet);
                  multChoice(patInd, patRet);
                      multChoice(legInd , legRet);
                      multChoice(slitInd , slitRet);
                      multChoice(pleatInd , pleatRet);
                      multChoice(lowerInd , lowerRet);
                      multChoice(brasInd , brasRet);
                      multChoice(occsnInd , occsnRet);
                      multChoice(styleInd , styleRet);
         }

          public void multChoice(int aInd, int aRet)
          {
        int aIndex = attribIndex[aInd];
        int aNumb = attribNumber[aInd];
                  SortAttribs[] sa = new SortAttribs[aNumb];
                      boolean vect1 = false;
        if (vectorNumb == 1)
        {
```

```
                        resultVector1[aInd] = new Vector();
                                vect1 = true;
        }
                            else
                            {
                        resultVector2[aInd] = new Vector();
                            }

                for (int i=0;i<aNumb ;i++ )
                    {
                            String fName = attribFullName[aInd].elementAt(i).toString();
//                              logger.debug(3," aInd "+aInd+" i: "+i+" fname "+fName);
                            float attf =
Float.parseFloat(dataRowVector[aIndex+i].elementAt(1).toString());
                            Arrays.fill(sa, i, i+1, new SortAttribs(attf, fName));
                    }
                        Arrays.sort(sa,new MyComparator());
                        for (int i=0;i<aNumb ;i++ )
                        {
//                          print(sa[i].toString());
                        }

        if (vect1)
                          resultVector1[aInd].add(new Integer(aRet));
        else
                          resultVector2[aInd].add(new Integer(aRet));

                        for (int i=0;i<attribNumber[aInd] ;i++)
                        {
          Vector v = new Vector();
                                v.insertElementAt(new Float(sa[i].getX()), 0);
                                String s = v.elementAt(0).toString();
                                s = s.substring(0, s.indexOf(".")+2);
                                if (vect1)
                                {
                                        resultVector1[aInd].add(sa[i].getName());
                                        resultVector1[aInd].add(s);

                                } else {
                              resultVector2[aInd].add(sa[i].getName());
                                resultVector2[aInd].add(s);
          }
        }
         }
         public void print(String s)
         {
      System.out.println(s);
         }

}
```

5) the CATEGORIZED OUTPUT -CHARACTERISTIC SEARCH ORDER (figure1- 26):

gtyp
unit

gocc	0
gsty	1
gcol	2
gnck	3
gslv	4
gftt	5
gftb	6
glnt	7
gleg	8
gpat	9
gtne	10
gmat	11
gslt	12
gplt	13
gbck	14
gclr	15
gbrs	16

6) the OUTPUT CHARACTERISTIC PASSING CRITERIA (figure1- 26):

gocc	3
gsty	3
gcol	3
gnck	3
gslv	3
gftt	9
gftb	9
glnt	5
gleg	5
gpat	9
gtne	9
gmat	9
gslt	5
gplt	5
gbck	5
gclr	9
gbrs	9

7) the SEARCHING SCHEMA/ RULES (figure 30):

```
/**
 * Version 3.0.6
 * Working and clean.
 * Works OK with servlets
 * Generated sorted arrays of all search scores.
 * Clean processLine() for exclude '9' feature
```

```
 */
// package guide;
//import SortAttribs;
//import MyComparator;
//import GetGarment;
import PictComparator;
import SortPicts;
import java.util.*;
import java.io.*;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;


public class GetScore5 {


   private static final int maxCol1 = 83;
   private static final int maxCol2 = 176;
   private static final int maxRow = 97;
   private static final int maxChoice1 = 12;
   private static final int maxChoice2 = 1;
   private static final int maxChoice = 13;

   public static final int maxAttrib = 17;

         private StringTokenizer st;

   private Vector[] choiceVector = new Vector[maxRow];

         public Vector[] resultVector1 = new Vector[maxAttrib];
         public Vector[] resultVector2 = new Vector[maxAttrib];

         private Vector attribFullName = null;

         public LineNumberReader lnReader = null;

         private int[] userChoice = new int[maxChoice];

         public int[] attribNumber = new int[] {4,5,4,6,9,2,15,4,16,9,5,3,3,4,2,6,11};
         private int[] attribIndex = new int[maxAttrib];

         private String switch1 = "no";
         private String switch2 = "no";
         private String switch3 = "allchoices";
         private String switch4 = "no";
   private int vectorNumb = 1;
         private int rowN = 0;

         private Vector colArray = null;

         int[] s9on = new int[maxAttrib];

   // files passed as a parameters
   protected String dataFile1 = null;
```

```
        private   PrintWriter out1;

        private int bothHit = 0;
        public Vector pictureVector = new Vector();

        public Properties props;

        public String[] manuf_arr;
        public String[] lname_arr;
        public String[] sname_arr;

        public int score_arr[];
        public int score1_arr[];

   private boolean twoLine = true;
   private boolean printAll = false;
        private Logger logger;
        private int garmentNumber1 = 0;
   private int garmentNumber2 = 0;

        public int start_p;
        public int end_p;
        public int aNumb;

        public SortPicts[] sp;

   /**
         * Constructor
         */
   GetScore5 (Properties props, int[] userChoice, int garmentNumber1,
            int garmentNumber2, Vector[] resultVector1, Vector[] resultVector2 /*,
PrintWriter out1*/) {
                    this.props = props;
                    this.userChoice = userChoice;
                    this.garmentNumber1 = garmentNumber1;
                    this.garmentNumber2 = garmentNumber2;
                    this.resultVector1 = resultVector1;
                    this.resultVector2 = resultVector2;
//                  this.out1 = out1;
        }

        /**
         * Find array of picture file names
         *
         */
        public void getPicture1()
        {

for (int i=0;i< userChoice.length ;i++)
{
//   print(""+userChoice[i]);
}
      if (props.getProperty("print.score","none").equals("all"))
      {
                    printAll = true;
      }
```

```
                getPictureData();

        getSorted(pictureVector);
}
        /**
         *  Parsing picture data file
         *
         */
        public void getPictureData()
        {
          LineNumberReader lnReader1;
          boolean good_data = true;
          int even_count = 0;
          int start_count=0;

           // open log file
      try
                {
                        // creating output files
                        String dir = props.getProperty("out.directory", "d:/temp");
                        String fl  = props.getProperty("out.file1", "Contest1.txt");
                        File f = new File(dir,fl);
                    out1 = new PrintWriter(
        new FileWriter(f));
      } catch(IOException e) {}

     // Get picture array start_stop

      if (garmentNumber2 == 0)
         twoLine = false;
      String garmCombo = ""+garmentNumber1+garmentNumber2;
           start_p =
           Integer.parseInt(props.getProperty("picture.start"+garmCombo,"-1"));
           end_p =
            Integer.parseInt(props.getProperty("picture.end"+garmCombo,"-1"));

//      logger.debug(168," garm "+garmCombo+" start "+start_p+" end "+end_p);
           if (start_p == Integer.parseInt("-1") || end_p == Integer.parseInt("-1"))
           {
                   good_data = false;
           }
           if (good_data)
           {
                        start_count = start_p-1;
                  // read picture data file
                        try
                        {
                          String dir = props.getProperty("garment.directory", "d:/temp");
                          File dataList1 = new File(dir,"pictdata.txt");
                          BufferedReader listIn1 = new BufferedReader(
                     new FileReader(dataList1));
                     lnReader1 = new LineNumberReader(listIn1);
                        }
                        catch(FileNotFoundException e)
                        { // Stream creation exception
```

```
                    System.err.println(e);
                    return;
                }
                    catch(IOException e)          // File read exception
                {
                    print("Error reading input file" + e );
                    return;
                    }

                String lineText = null;
                    boolean data_start = false;
                    boolean real_start = false;
                for (int i=0;i<1000 ;i++ )
                    {
                            try
                        {
                        lineText = lnReader1.readLine();
                            } catch (IOException e) {
                              System.out.println( "IO error reading data file. " +
e.toString() );
                                    System.exit(11);
                            }
                          if (lineText.indexOf("DATA_START") != -1)
                    {
                              data_start = true;
                                  continue;
                }
                            if (lineText == null)
                        {
                              break;
                            }
                    if (lineText.indexOf("DATA_END") != -1)
                    {
                                    break;
                            }
                      // remove 1 more line
                      if (data_start)
                      {
                        data_start = false;
                            real_start = true;
                            continue;
                      }
                      // real start
                      if (real_start)
                      {
                       if (start_count-- > 0)
            {
                        continue;
                       }
        if (even_count > end_p - start_p +1) {
                          break;
        }
                       if (twoLine)
                       {
                         if (even_count%2==0){
```

```
                                processLine(1,lineText);

                                        even_count++;
                                } else {
                                        processLine(2,lineText);

                                        even_count++;
                        }
                                } else {
                                  processLine(1,lineText);
                                        even_count++;
                                }

                        }
                }
        }
        }

        /**
         * Make real line for picture info processing
         *
         */
        private void processLine(int ln, String lineText)
        {

          int[] attrib_start = new int[] {/*gocc*/91,/*gsty*/97, /*gcol*/30, /*gnck*/4,
/*gslv*/13, /*gfft*/0,
                /*gftb*/85, /*glnt*/19, /*gleg*/74, /*gpat*/65, /*gtne*/45, /*gmat*/49,
/*gslt*/79, /*gplt*/82, /*gbck*/28, /*gclr*/9,
                /*gbrs*/89};

          int[] attrib_group = new int[] {/*gocc*/15,/*gsty*/16, /*gcol*/6, /*gnck*/1, /*gslv*/3,
/*gfft*/0,
                /*gftb*/13, /*glnt*/4, /*gleg*/10, /*gpat*/9, /*gtne*/7, /*gmat*/8, /*gslt*/11,
/*gplt*/12, /*gbck*/5, /*gclr*/2,
                /*gbrs*/14};

          int top_depth = 0;
          int bot_depth = 0;
          int count = 0;
          boolean both_depth = false;
          int tokenCount = 0;
          String[] tArray = new String[300];
          int minLength = Integer.parseInt(props.getProperty("pict.path.filter","1"));
          if (ln == 1)
          {
                StringTokenizer st = new StringTokenizer(lineText);
                bothHit = 0;
                while (st.hasMoreTokens())
                {
                        String token = st.nextToken();
                        tArray[tokenCount] = token;
                        tokenCount++;
                }
                // process this top line to the end
                top_depth = getDepth(1, attrib_start, attrib_group, tArray, resultVector1);
```

```
                    count++;
//                  both_depth = top_depth > 0;
            }
            else
            {
                    StringTokenizer st = new StringTokenizer(lineText);
               while (st.hasMoreTokens())
                    {
                            String token = st.nextToken();
                            tArray[tokenCount] = token;
                            tokenCount++;
                    }
                    // process this bottom line to the end
                    bot_depth = getDepth(2,attrib_start, attrib_group, tArray, resultVector2);
                    count++;
            }
            // process depth
            if (bothHit > minLength) {
              if (twoLine && ln == 2) {
                                    pictureVector.add(new Integer(bothHit));
                                            pictureVector.add(new Integer(top_depth));
                                            pictureVector.add(new String(tArray[20]));
                                            pictureVector.add(new String(tArray[21]+"L.jpg"));
                                            pictureVector.add(new
String(tArray[21]+"S.jpg"));

             } else if (!twoLine) {
                                            pictureVector.add(new Integer(bothHit));
                                            pictureVector.add(new Integer(top_depth));
                                            pictureVector.add(new String(tArray[20]));
                                            pictureVector.add(new String(tArray[21]+"L.jpg"));
                                            pictureVector.add(new
String(tArray[21]+"S.jpg"));
                        }

             }

            }

            /**
              *  Find how deep we can go
              *
              */
            private int getDepth(int layer, int[] st_arr, int[] gr_arr, String[] t_arr, Vector[]
resultVector)
   {
      int passed = 0;
              int thisHit = 0;
              // process top
              int i = 3, j = 0;
              String[] nameArr;
      for (; i<maxAttrib+3 ;i++,j++ ) {
                // get token
                String tk = t_arr[i];
       if (tk.length() == 1) { // single character
                    int attrIndx = Integer.parseInt(props.getProperty("attrib.numb"+tk,"-1"));
```

```
                    if (attrIndx >= 0) {
                        int indx = st_arr[j]+attrIndx;
                        nameArr = new String[1];
                                nameArr[0] =
props.getProperty("garment.attrib"+indx,"bad");

                    }
        else
                                continue;
      }
                else {  // combinqation of characters
                        nameArr = new String[tk.length()];
                        for (int k=0;k<tk.length() ;k++ ) {
                           int indx = st_arr[j]+Integer.parseInt(props.getProperty(gr_arr[j]+
                                "."+tk+"."+k,"-1"));

                                nameArr[k] = props.getProperty("garment.attrib"+indx,
"nonsense");
                        }
                }
             passed = getCompare(nameArr, gr_arr[j], resultVector[gr_arr[j]], t_arr[21], j);
                if (passed > 0) {
                        thisHit++;
                                bothHit++;
                } else if (thisHit < 2) {
                        break;

                }
                else if (passed < 0)      {
            thisHit = -15;
                                bothHit = -15;
                                break;
                }

    }
    if (layer == 1) {
                        out1.println(" Was file hit on top for file "+t_arr[21]+" Hits
"+thisHit+" j: "+j);
          } else {
                        out1.println(" Was file hit on bottom for file "+t_arr[21]+" Hits
"+thisHit+" j: "+j);
          }
           out1.flush();
     return thisHit;
  }

          /**
           * Make comparison. if not go try attribute
           * with next best score till score < 2.5
           *
           */
           private int getCompare(String[] nArray, int aIndx, Vector scoreVector, String file,
int col)
           {
         Arrays.sort(nArray);
                    String s9 = props.getProperty("pict.score.9","off");
```

```
                    for (int k=0;k < maxAttrib ;k++ )
                    {
                      s9on[k] = Integer.parseInt(props.getProperty("pict.on9."+k,"0"));
                    }
                    boolean sw9 = s9.equals("on");
                    for (int i=0;i<nArray.length ;i++ )
                    {
                     out1.println(" Array nArray for file "+file+" at i: "+i+" is "+nArray[i]);
                    }
                    int pos = -1;
        // first try to test if score '9' is presense.
                    // If so no other testing
                    for (int i=0,j = 1;i<(scoreVector.size()-1)/2 ;i++,j++ )
                    {
                        String attrName = scoreVector.elementAt(2*i+1).toString();
                                double attrScore =
Double.parseDouble(scoreVector.elementAt(2*i+2).toString());
                    if (attrScore == 9.0)
                                {
                                   pos = Arrays.binarySearch(nArray,attrName);
 out1.println("Get Compare in '9' for file "+file+" attrName "+attrName+" Score
"+attrScore+" pos "+pos);
 out1.flush();
                                                if (pos >= 0 && sw9 && s9on[col] == 1)
                                                {
 out1.println("I should quit here"+ "sw9: "+s9on[col]+ "pos: "+pos);
 out1.flush();
 return -1;

                                                }
                                }
                    }

                    // try next in score
                    for (int i=0,j = 1;i<(scoreVector.size()-1)/2 ;i++,j++ )
                    {
                        String attrName = scoreVector.elementAt(2*i+1).toString();
                                double attrScore =
Double.parseDouble(scoreVector.elementAt(2*i+2).toString());
            double pFilt = Double.parseDouble(props.getProperty("pict.filter","2.0"));
                                if (attrScore < pFilt)
                                {
                                   pos = Arrays.binarySearch(nArray,attrName);
 out1.println("Get Compare: for file "+file+" attrName "+attrName+" Score "+attrScore+"
pos "+pos);
 out1.flush();
                                                if (pos >= 0)
                                                {
                                                  return 1;
                                                }
                                }
                    }
                    if (pos < 0)
                    {
                      return 0;
```

```
                }
                return 1;
    }

     private boolean matchStyle(String styleBloc, int choice)
          {
          String ch = props.getProperty("guides.col"+choice,"none");
                         String[] styleNames = new String[styleBloc.length()];
                         for (int k=0;k<styleBloc.length() ;k++ )
                         {
                            int indx = 64+Integer.parseInt(props.getProperty("16."+
                                        styleBloc+"."+k,"-1"));

                                    styleNames[k] = props.getProperty("guides.col"+indx,
"nonsense");
                                    if (styleNames[k].equals(ch))
                                        return true;
          }
                         return false;
          }

          /**
           * Sort Array of objects
           *
           */
           public void getSorted(Vector pVector)
           {
                         aNumb = pVector.size()/5;
                    sp = new SortPicts[aNumb];
               for (int i=0;i<aNumb ;i++ )
                    {
                         int score = Integer.parseInt(pVector.elementAt(i*5).toString());
            int score1 = Integer.parseInt(pVector.elementAt(i*5+1).toString());
                         String manuf = pVector.elementAt(i*5+2).toString();
                         String lname = pVector.elementAt(i*5+3).toString();
                         String sname = pVector.elementAt(i*5+4).toString();
                         Arrays.fill(sp, i, i+1, new
SortPicts(score,score1,manuf,lname,sname));
                    }
                         Arrays.sort(sp,new PictComparator());
                         for (int i=0;i<aNumb ;i++ )
                         {
                            out1.println(sp[i].toString());
                                 out1.flush();
                         }

               int div = (int)(aNumb/5);
                         int hm = div * 5;
                         if (hm < aNumb)
                         {
                           hm += 5;
                         }
//          logger.debug(526," Pictures Number "+aNumb+" to whole Set  "+hm);
                         score_arr = new int[hm];
          score1_arr = new int[hm];
                         manuf_arr = new String[hm];
```

```
                        lname_arr = new String[hm];
                        sname_arr = new String[hm];
                        for (int i=0;i<aNumb ;i++)
                        {
                           score_arr[i] = sp[i].getX();
                                score1_arr[i] = sp[i].getTop();
                                manuf_arr[i] = sp[i].getManuf();
                                lname_arr[i] = sp[i].getLname();
                                sname_arr[i] = sp[i].getSname();
                        }
        for (int i=aNumb;i<hm ;i++)
                        {
                           score_arr[i] = score_arr[aNumb-1];
                           score1_arr[i] = score1_arr[aNumb-1];
                                manuf_arr[i] = manuf_arr[aNumb-1];
                                lname_arr[i] = lname_arr[aNumb-1];
                                sname_arr[i] = sname_arr[aNumb-1];
                        }

                        for (int i=0;i<hm ;i++)
                        {
                                out1.println(" i:"+i+" both_score "+score_arr[i]+" top_score
"+score1_arr[i]+" manufact "
                                +manuf_arr[i]+" lname "+lname_arr[i]);
                                out1.flush();
                        }
         }


// ROW gtyp     unit      gocc/*91*/ gcol/*30*/ gfft /*0*/ gftb/*89*/ glnt/*19*/
// gleg/*74*/ gnck/*4*/ gslv/*13*/ gpat/*65*/ gtne/*45*/ gmat/*49*/ gslt/*79*/
// gplt/*82*/ gbck/*28*/ gclr/*9*/ gbrs/*89*/ gsty    manf     filename



         public void print(String s)
         {
      System.out.println(s);
         }

}
```

8) the CHARACTERIZED INVENTORY DATABASE(figure 34):

gtyp (garment combo type): 13-shirt/skirt 14-shirt/pant 15-shirt-shorts 2x-dress 26-jacket/dress 36-jacket/skirt 46-jacket/pant 56-jacket/shorts
unit (garment unit): u1-shirts u2-dresses u3-skirts u4-pants u5-shorts u6-jackets
gocc (garment occasion): 0-formal 1-semiformal 2-dressy 3-casual 4-party 5-business
gcol (garment color): 0-black 1-blue 2-brown 3-gold 4-green 5-grey 6-indigo 7-orange 8-pink 9-red a-silver b-tan c-violet d-white e-yellow
gftt (garment fit upper body): 0-loose 1-normal 2-fitted 3-tight
gftb (garment fit bottom): 0-loose 1-normal 2-fitted 3-tight
glnt (garment length[height]):0-crop 1-waist 2-hip 3-thigh 4-knee 5-calve 6-ankle 7-heel 8-floor
gleg (garment leg): 0-bell 1-straight 2-tapered 3-wide 4-boot cut
gnck (garment neck): 0-neck concealing 1-neck lined 2-v-neck/scoop neck 3-low cut 4-neck-less
gslv (garment sleeve):0-sleeveless 1- thin strap 2-shoudler sleeve 3-short sleeve 4-3/4 sleeve 5-long sleeve
gpat (garment pattern): 0-solid 1-plad/checker 2-v-stripe 3-h-stripe 4-animal print 5-paisley 6-polka dots 7-floral 8-other
gtne (garment color tone): 0-light 1-bold 2-rich 3-dark
gmat (garment material): 0-animal hide 1-denim 2-cashmere 3-cotton 4-wool 5-acetate 6-polyester 7-spandex 8-beaded 9-rayon a-silk b-lace c-linen d-lycra e-satin f-other
gslt (garment slit [skirt/dress]): 0-none 1-none revealing slit 2-revelaing slit
gplt (garment pleat):0-flat front 1-single pleat 2-double pleat
gbck (garment back): 0-revealing 1-concealing
gclr (garment collar/lapel): 0-pointed 1-rounded 2-band 3-none
gbrs (garment breast): 0-single breasted 1-double breasted
gsty (garment style): 0-casual 1-revealing 2-grunge 3-ethnic 4-retro5-urban 6-savvy 7-sporty 8-trendy 9-conservative a-flashy
manf (manufacture code): ##
file (filename) : ###

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	Glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
1	13	1	24	2	68	12	5	2	X	2	X	6	0	3B	X	X	1	3	X	10	151
2	13	3	24	2	68	X	X	X	1	5	1	6	0	3	0	0	X	X	X	10	151
3	13	1	124	2	8	2	5	2	X	1	X	0	13	3	X	X	1	3	X	10	152
4	13	3	124	2	8	X	X	X	0	6	1	0	13	3	1	2	X	X	X	10	152
5	13	1	234	2	893	2	5	2	X	2	X	8	12	9F	X	X	1	3	X	10	153
6	13	3	234	2	190	X	X	X	1	5	1	6	13	83	0	0	X	X	X	10	153
7	13	1	124	A	149	1	3	1	X	2	X	8	12	68	X	X	0	3	X	11	154
8	13	3	124	A	1	X	X	X	1	5	1	0	0	3	1	0	X	X	X	11	154
9	13	1	12	69	9	1	4	2	X	1	X	0	23	3	X	X	0	2	X	12	155
10	13	3	12	69	9	X	X	X	2	4	2	0	1	E	0	0	X	X	X	12	155
11	13	1	12	69	1	1	5	2	X	1	X	0	23	3	X	X	0	2	X	12	156
12	13	3	12	69	1	X	X	X	2	4	2	0	1	E	0	0	X	X	X	12	156
13	13	1	1	69	6	1	5	2	X	1	X	0	23	3	X	X	0	2	X	12	157
14	13	3	1	69	6	X	X	X	2	4	2	0	1	5	0	0	X	X	X	12	157
15	13	1	234	69A	BE	0	5	2	X	2	X	8	12	3D	X	X	1	3	X	15	158
16	13	3	234	69A	B	X	X	X	1	4	1	0	0	3	0	0	X	X	X	15	158
17	13	1	234	56	0	1	0	2	X	2	X	0	123	39	X	X	1	3	X	15	159
18	13	3	234	56	0D	X	X	X	2	4	1	6	0	39	0	0	X	X	X	15	159
19	13	1	24	69	0D	0	5	2	X	1	X	8	0	3	X	X	1	3	X	15	160
20	13	3	24	69	0D	X	X	X	1	4	1	8	0	3	0	0	X	X	X	15	160
21	13	1	2345	689	4	0	5	2	X	1	X	0	1	3	X	X	1	3	X	15	161
22	13	3	2345	689	4	X	X	X	2	4	1	23	13	3	0	0	X	X	X	15	161
23	13	1	2345	789	1	0	5	2	X	2	X	0	1	3	X	X	1	3	X	16	162
24	13	3	2345	789	0	X	X	X	2	4	2	0	123	36	1	0	X	X	X	16	162
25	13	1	25	69	2	0	5	23	X	1	X	0	23	4	X	X	1	3	X	16	163
26	13	3	25	69	2	X	X	X	2	4	2	0	23	4	0	0	X	X	X	16	163
27	13	1	124	1A	0A	3	12	2	X	1	X	8	13	5B	X	X	0	3	X	17	164
28	13	3	124	1A	26	X	X	X	12	5	2	0	23	4	0	0	X	X	X	17	164
29	13	1	34	58	8C	3	1	23	X	1	X	0	0	5	X	X	0	3	X	18	165
30	13	3	34	58	C	X	X	X	12	4	1	0	23	F	0	0	X	X	X	18	165

31-60
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	Glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
31	13	1	124	69	0	1	1	12	X	1	X	0	123	3	X	X	1	3	X	1	166
32	13	3	124	69	0	X	X	X	3	56	2	0	123	3	0	0	X	X	X	1	166
33	13	1	124	8A	1	0	2	1	X	1	X	7	13	36	X	X	1	3	X	21	167
34	13	3	124	8A	1	X	X	X	3	5	2	0	1	D	0	0	X	X	X	21	167
35	13	1	1	9	6	4	1	1	X	1	X	8	2	B	X	X	1	3	X	21	168
36	13	3	1	9	6	X	X	X	1	7	1	0	23	3A	0	0	X	X	X	21	168
37	13	1	1	9	EB6	4	1	1	X	1	X	18	2	5	X	X	1	3	X	21	169
38	13	3	1	9	6	X	X	X	1	7	1	1	1	5	0	0	X	X	X	21	169
39	13	1	1	6	0A	4	1	2	X	1	X	38	13	9	X	X	1	3	X	21	170
40	13	3	1	6	0A	X	X	X	12	7	1	38	13	9	0	0	X	X	X	21	170
41	13	1	4	78A	8C	4	0	3	X	0	X	0	1	5	X	X	1	3	X	23	171
42	13	3	4	78A	6	X	X	X	2	3	2	0	3	F	0	0	X	X	X	23	171
43	13	1	2	68A	4	0	4	2	X	1	X	8	0	5A	X	X	1	3	X	25	172
44	13	3	2	68A	4	X	X	X	1	4	1	0	0	3	0	0	X	X	X	25	172
45	13	1	14	18A	0	3	1	2	X	0	X	0	3	F	X	X	1	3	X	27	173
46	13	3	14	18A	C	X	X	X	1	4	1	6	1	5	0	0	X	X	X	27	173
47	13	1	23	79	0	1	5	2	X	1	X	0	3	2	X	X	1	3	X	29	174
48	13	3	23	79	0	X	X	X	1	4	1	0	3	2	0	0	X	X	X	29	174
49	13	1	25	68	27	3	3	2	X	1	X	8	2	C	X	X	1	3	X	2	175
50	13	3	25	68	79	X	X	X	1	4	1	0	2	9	0	0	X	X	X	2	175
51	13	1	25	68	27	1	5	2	X	1	X	8	2	36	X	X	0	3	X	2	176
52	13	3	25	68	8	X	X	X	1	4	1	0	1	0	0	0	X	X	X	2	176
53	13	1	34	56	5	0	1	2	X	1	X	8	3	4	X	X	1	3	X	31	177
54	13	3	34	56	5	X	X	X	2	4	2	8	3	9	0	0	X	X	X	31	177
55	13	1	12	9	0	2	2	1	X	2	X	0	3	9	X	X	1	3	X	38	178
56	13	3	12	9	0	X	X	X	2	6	1	0	3	9	0	0	X	X	X	38	178
57	13	1	12	6	0	1	5	1	X	1	X	0	3	9	X	X	1	3	X	38	179
58	13	3	12	6	0	X	X	X	1	6	1	0	3	9	0	0	X	X	X	38	179
59	13	1	12	69	0	0	5	2	X	1.	X	0	123	4	X	X	1	3	X	3	180
60	13	3	12	69	B	X	X	X	2	4	2	0	0	E	0	0	X	X	X	3	180

61-89
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	GInt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
61	13	1	234	68	0	0	5	1	X	2	X	0	123	4	X	X	1	3	X	41	181
62	13	3	234	68	B	X	X	X	2	4	2	4	1	0	0	0	X	X	X	41	181
63	13	1	1	9	3	1	2	1	X	1	X	8	13	5	X	X	1	3	X	41	182
64	13	3	1	9	0	X	X	X	2	7	12	0	123	3	0	0	X	X	X	41	182
65	13	1	23	69	0	0	5	2	X	1	X	0	123	4	X	X	1	3	X	5	183
66	13	3	23	69	B	X	X	X	1	5	1	3	0	3	0	0	X	X	X	5	183
67	13	1	234	8	4	0	5	1	X	1	X	0	3	4	X	X	1	3	X	7	184
68	13	3	234	8	4	X	X	X	1	5	1	8	13	6	0	0	X	X	X	7	184
69	13	1	34	8	9	4	0	2	X	1	X	0	2	3	X	X	1	3	X	7	185
70	13	3	34	8	E	X	X	X	0	4	1	0	2	3	2	0	X	X	X	7	185
71	13	1	4	5	2B	2	5	1	X	1	X	8	2	5	X	X	1	0	X	11	186
72	13	3	4	5	0B	X	X	X	1	4	2	1	23	4	0	0	X	X	X	11	186
73	13	1	1	69	4C	3	0	2	X	1	X	0	23	5	X	X	1	3	X	50	187
74	13	3	1	69	4C	X	X	X	0	8	1	0	23	5	0	1	X	X	X	50	187
75	13	1	2	3	0	0	5	3	X	1	X	0	3	4	X	X	1	3	X	4	411
76	13	3	2	3	0	X	X	X	3	6	2	0	3	5	0	0	X	X	X	4	411
77	13	1	24	3	0	1	4	2	X	0	X	0	3	AD	X	X	1	3	X	4	412
78	13	3	24	3	1	X	X	X	0	7	0	8	1	AD	0	0	X	X	X	4	412
79	13	1	12	3	0	2	5	1	X	1	X	6	3	5	X	X	1	0	0	4	414
80	13	3	12	4	0	X	X	X	0	7	1	6	3	5A	0	0	X	X	X	4	414
81	13	1	4	3	6	1	25	2	X	1	X	5	2	3	X	X	1	3	X	63	417
82	13	3	4	3	6	X	X	X	1	3	2	5	2	3	0	0	X	X	X	63	417
83	13	1	4	3	6	0	2	2	X	1	X	5	2	3	X	X	0	3	X	63	418
84	13	3	4	3	6	X	X	X	2	3	2	0	2	3	2	0	X	X	X	63	418
85	13	1	34	5	0	0	5	2	X	1	X	0	123	4	X	X	1	3	X	28	420
86	13	3	34	5	0	X	X	X	1	4	2	0	123	3	0	0	X	X	X	28	420
87	13	1	234	3	0	0	5	2	X	1	X	0	123	3	X	X	1	3	X	71	428
88	13	3	234	3	9	X	X	X	1	5	1	8	13	3	1	0	X	X	X	71	428
89	13	1	1	78	5	1	3	2	X	1	X	0	0	4	X	X	1	3	X	72	431

90-120
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
90	13	3	1	78	A	X	X	X	2	8	1	0	1	5	0	0	X	X	X	72	431
91	13	1	14	5	0	2	4	3	X	1	X	0	123	0	X	X	1	0	X	72	433
92	13	3	14	5	9C	X	X	X	3	4	2	0	1	0	0	0	X	X	X	72	433
93	13	1	12	69	B	1	5	2	X	1	X	0	0	5	X	X	1	3	X	72	434
94	13	3	12	69	B	X	X	X	2	4	1	0	1	5	0	0	X	X	X	72	434
95	13	1	0	1	0	3	5	2	X	2	X	0	123	5A	X	X	1	0	X	56	452
96	13	3	0	1	0	X	X	X	1	7	1	0	123	5A	1	0	X	X	X	56	452
97	13	1	0	16	5	1	2	3	X	1	X	0	0	8	X	X	1	3	X	73	453
98	13	3	0	16	5	X	X	X	1	7	2	0	0	D	2	0	X	X	X	73	453
99	13	1	0	5	0	3	5	3	X	1	X	0	123	E	X	X	1	3	X	20	465
100	13	3	0	5	0	X	X	X	3	7	0	0	123	3D	2	0	X	X	X	20	465
101	13	1	0	5	0	24	2	3	X	1	X	0	123	E	X	X	1	3	X	20	466
102	13	3	0	5	0	X	X	X	2	7	0	0	123	3D	1	0	X	X	X	20	466
103	14	1	124	168A	BA	1	0	2	X	0	X	8	0	F	X	X	1	3	X	11	251
104	14	4	124	168A	B	X	X	X	0	8	0	0	0	95	X	2	X	X	X	11	251
105	14	1	124	8A	9	0	0	1	X	1	X	0	2	9	X	X	1	2	X	11	252
106	14	4	124	8A	149	X	X	X	0	7	3	3	1	9	X	0	X	X	X	11	252
107	14	1	234	168A	3	0	0	3	X	1	X	0	2	0	X	X	1	2	X	13	253
108	14	4	234	168A	23	X	X	X	2	7	3	1	2	607	X	0	X	X	X	13	253
109	14	1	234	068A	0	0	5	12	X	1	X	0	123	367	X	X	1	2	X	15	254
110	14	4	234	068A	0D	X	X	X	2	56	1	6	0	367	X	0	X	X	X	15	254
111	14	1	235	69	235B	1	5	12	X	1	X	5	2	59A	X	X	1	3	X	16	255
112	14	4	235	69	5B	X	X	X	12	7	1	1	2	65	X	0	X	X	X	16	255
113	14	1	234	168A	230	23	5	2	X	1	X	4	2	F	X	X	1	3	X	16	256
114	14	4	234	168A	2	X	X	X	12	7	4	0	3	F	X	0	X	X	X	16	256
115	14	1	12	1689A	0	3	0	1	X	1	X	0	123	8	X	X	0	3	X	16	257
116	14	4	12	1689A	0	X	X	X	1	7	1	0	123	9	X	0	X	X	X	16	257
117	14	1	12	16A8	A	3	0	1	X	1	X	0	1	8	X	X	1	3	X	17	258
118	14	4	12	168A	5	X	X	X	1	7	1	0	1	45	X	0	X	X	X	17	258
119	14	1	34	5	024B	3	0	3	X	0	X	8	3	3	X	X	0	3	X	18	259
120	14	4	34	5	2	X	X	X	2	45	4	0	3	37	X	0	X	X	X	18	259

121-152
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
121	14	1	3	08A	4	2	2	2	X	1	X	0	1	37	X	X	1	3	X	1	260
122	14	4	3	08A	B	X	X	X	0	7	4	0	0	3	X	0	X	X	X	1	260
123	14	1	23	1689	24	23	0	2	X	1	X	8	3	59	X	X	1	3	X	21	261
124	14	4	23	1689	24	X	X	X	1	7	1	1	3	63	X	0	X	X	X	21	261
125	14	1	12	689	1	1	0	1	X	1	X	2	3	59	X	X	0	3	X	21	262
126	14	4	12	689	1	X	X	X	0	8	3	0	0	5F	X	0	X	X	X	21	262
127	14	1	34	48A	2	2	5	3	X	2	X	0	2	0	X	X	1	0	X	22	263
128	14	4	34	48A	0	X	X	X	3	6	1	0	123	0	X	0	X	X	X	22	263
129	14	1	34	2458	3CE	0	5	3	X	1	X	3	2	379	X	X	1	2	X	22	264
130	14	4	34	2458	1C	X	X	X	2	7	1	8	3	10	X	0	X	X	X	22	264
131	14	1	234	018A	43	4	0	3	X	0	X	8	0	37	X	X	0	3	X	23	265
132	14	4	234	018A	4	X	X	X	1	7	4	0	0	39	X	0	X	X	X	23	265
133	14	1	4	5	4	4	0	3	X	0	X	8	0	58	X	X	0	3	X	23	266
134	14	4	4	5	4B	X	X	X	1	7	1	0	0	3	X	0	X	X	X	23	266
135	14	1	234	8	C	23	0	2	X	1	X	0	3	9	X	X	1	3	X	23	267
136	14	4	234	8	1	X	X	X	1	7	4	0	0	3	X	0	X	X	X	23	267
137	14	1	34	28	D	1	0	3	X	1	X	0	0	67	X	X	1	3	X	23	268
138	14	4	34	28	0	X	X	X	1	1	4	0	123	0	X	0	X	X	X	23	268
139	14	1	14	25	0	4	5	1	X	1	X	0	123	D	X	X	1	3	X	23	269
140	14	4	14	25	0	X	X	X	1	7	3	0	123	D	X	0	X	X	X	23	269
141	14	1	3	0	B	0	5	2	X	1	X	0	3	4	X	X	1	3	X	23	270
142	14	4	3	0	1	X	X	X	2	7	1	0	0	6	X	0	X	X	X	23	270
143	14	1	1234	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	X	23	271
144	14	4	1234	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	X	23	271
145	14	1	34	69	9B	2	5	2	X	2	X	0	0	C	X	X	1	0	X	24	272
146	14	4	34	69	0	X	X	X	2	7	4	2	123	6	X	0	X	X	X	24	272
147	14	1	34	248	2	2	1	3	X	1	X	0	2	5	X	X	1	3	X	24	273
148	14	4	34	248	8	X	X	X	2	8	4	1	1	6	X	0	X	X	X	24	273
149	14	1	14	1	2	4	1	2	X	3	X	0	2	5B	X	X	1	3	X	25	274
150	14	4	14	1	2	X	X	X	2	7	1	0	2	8	X	0	X	X	X	25	274
151	14	1	124	48	1CD	0	1	2	X	1	X	8	1	5	X	X	1	3	X	26	275
152	14	4	124	48	C	X	X	X	2	7	3	0	1	8	X	0	X	X	X	26	275

153 to 183
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
153	14	1	124	48	0D6	1	2	1	X	1	X	8	1	5	X	X	1	3	X	26	276
154	14	4	124	48	A	X	X	X	1	8	1	0	1	5	X	0	X	X	X	26	276
155	14	1	235	69	5	3	5	2	X	1	X	0	3	5	X	X	1	0	X	27	277
156	14	4	235	69	5	X	X	X	1	7	4	0	0	5	X	0	X	X	X	27	277
157	14	1	234	78	05D	1	5	3	X	1	X	8	1	3	X	X	1	3	X	27	278
158	14	4	234	78	5	X	X	X	2	7	4	0	3	3	X	0	X	X	X	27	278
159	14	1	12	68	19	4	2	1	X	1	X	8	1	F	X	X	1	3	X	27	279
160	14	4	12	68	19	X	X	X	3	7	4	8	1	F	X	0	X	X	X	27	279
161	14	1	12	9	D	1	5	2	X	1	X	0	0	3	X	X	1	0	X	32	280
162	14	4	12	9	0	X	X	X	2	7	4	0	123	F	X	0	X	X	X	32	280
163	14	1	25	69	0	0	5	1	X	2	X	0	123	9	X	X	1	3	X	34	281
164	14	4	25	69	0	X	X	X	2	7	1	0	123	9	X	0	X	X	X	34	281
165	14	1	2345	69	0	1	5	3	X	1	X	0	123	3	X	X	1	3	X	41	282
166	14	4	2345	69	0	X	X	X	1	7	1	0	123	3	X	0	X	X	X	41	282
167	14	1	234	9	0	1	4	2	X	2	X	0	123	5	X	X	1	3	X	5	283
168	14	4	234	9	0	X	X	X	1	8	1	0	123	5	X	0	X	X	X	5	283
169	14	1	234	69	0	0	1	2	X	1	X	0	123	3	X	X	1	3	X	6	284
170	14	4	234	69	0	X	X	X	1	7	1	0	123	3	X	0	X	X	X	6	284
171	14	1	34	35	0	1	3	0	X	0	X	0	123	3	X	X	1	3	X	6	285
172	14	4	34	35	0	X	X	X	1	8	2	0	123	3	X	0	X	X	X	6	285
173	14	1	125	9	0	0	5	2	X	1	X	0	123	3	X	X	1	3	X	9	286
174	14	4	125	9	0	X	X	X	2	7	4	0	123	3	X	0	X	X	X	9	286
175	14	1	4	4A	0D83	3	1	1	X	1	X	2	1	6	X	X	1	0	X	52	287
176	14	4	4	4A	0D83	X	X	X	2	7	1	2	1	6	X	0	X	X	X	52	287
177	14	1	4	18A	A	1	1	2	X	0	X	0	1	5	X	X	0	3	X	42	419
178	14	4	4	18A	0	X	X	X	3	7	0	0	123	3	X	0	X	X	X	42	419
179	14	1	1	6	0	4	5	2	X	1	X	0	123	5	X	X	1	3	X	50	421
180	14	4	1	6	0	X	X	X	1	7	1	0	123	3	X	0	X	X	X	50	421
181	14	1	1	568	0	4	0	3	X	1	X	0	123	5	X	X	1	3	X	50	423
182	14	4	1	568	0	X	X	X	2	7	1	0	123	5	X	0	X	X	X	50	423
183	14	1	1	2	0	2	1	2	X	3	X	0	123	3	X	X	1	3	X	10	425

184-213
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
184	14	4	1	2	0	X	X	X	0	6	3	0	123	3	X	0	X	X	X	10	425
185	14	1	1	46	B	1	4	1	X	1	X	0	1	8	X	X	1	3	X	72	432
186	14	4	1	46	B	X	X	X	2	7	1	0	0	3	X	0	X	X	X	72	432
187	14	1	4	156A	2	4	0	3	X	0	X	3	1	F	X	X	1	3	X	66	435
188	14	4	4	156A	0	X	X	X	2	7	1	0	123	3	X	0	X	X	X	66	435
189	14	1	4	5	4	3	4	3	X	1	X	0	0	8	X	X	1	3	X	66	436
190	14	4	4	5	0	X	X	X	2	7	1	0	123	3	X	0	X	X	X	66	436
191	14	1	234	5	0	3	5	2	X	2	X	0	3	3	X	X	1	3	X	73	437
192	14	4	234	5	2	X	X	X	2	7	4	3	3	0	X	0	X	X	X	73	437
193	14	1	14	5	0D	2	2	2	X	1	X	0	13	2	X	X	1	3	X	73	438
194	14	4	14	5	0D	X	X	X	3	7	4	3	1	0	X	0	X	X	X	73	438
195	14	1	234	45	09D	0	2	0	X	4	X	8	2	3	X	X	1	3	X	74	443
196	14	4	234	45	D	X	X	X	1	7	1	0	0	3	X	0	X	X	X	74	443
197	14	1	4	35	2	0	5	2	X	0	X	0	23	56DE	X	X	1	3	X	66	450
198	14	4	4	35	2B	X	X	X	2	7	4	0	23	56DE	X	0	X	X	X	66	450
199	14	1	0	67	0	1	2	2	X	1	X	0	123	8D	X	X	1	3	X	32	461
200	14	4	0	67	0	X	X	X	2	7	4	0	123	8D	X	0	X	X	X	32	461
201	14	1	14	5	24	4	1	2	X	1	X	8	23	8	X	X	1	3	X	82	464
202	14	4	14	5	23	X	X	X	3	7	4	0	23	5	X	0	X	X	X	82	464
203	14	1	0	5	0	34	0	3	X	1	X	0	123	E	X	X	1	3	X	20	467
204	14	4	0	5	0	X	X	X	0	7	3	0	123	3D	X	0	X	X	X	20	467
205	14	1	4	578A	0D	4	2	2	X	23	X	4	123	0D	X	X	0	3	X	63	469
206	14	4	4	578A	9	X	X	X	2	7	4	0	12	0D	X	0	X	X	X	63	469
207	15	1	24	138A	3	1	0	2	X	1	X	0	3	5E	X	X	1	3	X	23	354
208	15	5	24	138A	9	X	X	X	3	23	1	0	3	###	0	0	X	X	X	23	354
209	15	1	34	168	1	2	5	3	X	1	X	0	0	37	X	X	1	3	X	65	355
210	15	5	24	168	1	X	X	X	2	23	1	0	0	0	0	0	X	X	X	65	355
211	15	1	34	01A	B	3	3	0	X	1	X	0	0	3	X	X	1	0	X	10	356
212	15	5	34	01A	B	X	X	X	1	23	1	0	0	3	0	0	X	X	X	10	356
213	15	1	24	18A	19D	4	0	2	X	1	X	8	0	937	X	X	1	3	X	66	358

214-245
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
214	15	5	24	18A	D	X	X	X	1	2	1	0	0	3	0	0	X	X	X	66	358
215	15	1	4	1A	D	4	0	3	X	0	X	0	0	7	X	X	0	3	X	67	361
216	15	5	4	1A	0	X	X	X	2	23	1	0	3	95	0	0	X	X	X	67	361
217	15	1	34	18A	389	1	0	3	X	1	X	7	1	59	X	X	1	3	X	32	362
218	15	5	34	18A	0	X	X	X	3	23	1	0	3	0	0	0	X	X	X	32	362
219	15	1	24	18A	7	0	5	2	X	1	X	0	1	9	X	X	1	2	X	32	363
220	15	5	24	18A	789D	X	X	X	2	23	1	8	1	3	0	0	X	X	X	32	363
221	15	1	34	58	B	0	5	0	X	1	X	0	0	24	X	X	1	3	X	78	451
222	15	5	34	58	2	X	X	X	2	2	1	0	3	4	X	0	X	X	X	78	451
223	20	2	24	678	E	1	2	2	1	4	2	0	0	4	0	X	1	3	X	11	1
224	20	2	2	2	09B	1	5	2	1	6	1	8	123	56	0	X	1	3	X	10	2
225	20	2	4	2	1	2	3	3	0	3	0	7	12	6	0	X	1	3	X	10	3
226	20	2	12	2	D	1	5	2	0	5	2	0	0	3	0	X	1	2	X	10	4
227	20	2	34	2	09D	2	5	2	0	3	0	7	3	3	0	X	1	0	X	10	5
228	20	2	12	2	0	0	5	3	2	5	1	0	123	E	0	X	1	3	X	10	6
229	20	2	34	0	1B9E	0	5	1	0	4	1	2	1	3	0	X	1	3	X	11	7
230	20	2	234	678	2B	2	5	1	1	4	1	0	2	0	0	X	1	0	X	13	8
231	20	2	25	9	0B	0	5	2	1	5	1	8	2	4	1	X	1	2	X	16	9
232	20	2	12	9	29	2	1	2	1	5	1	0	2	5A	0	X	1	3	X	17	10
233	20	2	2	78	8C	2	2	2	2	4	2	23	1	6	0	X	1	3	X	26	11
234	20	2	25	56	0D	2	4	2	2	5	2	2	3	3	0	X	1	0	X	8	12
235	20	2	0	2	4	4	1	1	0	6	1	0	0	8	0	X	1	X	X	5	13
236	20	2	12	29	0	1	5	2	1	5	1	0	123	E	0	1	1	0	X	5	14
237	20	2	34	78A	8B	3	1	2	1	3	1	8	1	8E	0	X	1	3	X	4	15
238	20	2	1	38	3	4	1	2	2	4	1	0	D5	3	0	X	1	3	X	4	16
239	20	2	25	9	0	2	2	1	1	5	1	0	123	35	0	X	1	3	X	41	17
240	20	2	0	136	237	4	0	2	2	6	2	8	2	5	0	X	1	X	X	4	18
241	20	2	.125	68	7	1	2	2	1	4	1	0	2	5D	0	X	1	3	X	2	19
242	20	2	2	69	9	1	2	2	2	4	1	0	1	45	0	X	1	3	X	2	20
243	20	2	14	136A	0	3	4	2	1	4	1	8	123	35	0	X	1	0	X	2	21
244	20	2	21	689	4	2	2	2	1	4	1	0	12	45	0	X	1	3	X	2	22
245	20	2	24	1A	284	0	2	3	1	4	1	7	2	35	0	X	1	2	X	25	23

246-275
DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
246	20	2	1	45A	9	0	0	1	2	4	2	0	3	8	0	X	0	2	X	11	24
247	20	2	12	12	0	2	2	2	1	5	2	0	123	E	0	X	1	3	X	55	25
248	20	2	24	048A	26	2	2	2	1	4	2	8	13	A	0	X	1	3	X	57	26
249	20	2	0	16A	0	3	5	2	2	8	2	0	123	69	2	X	1	X	X	60	27
250	20	2	14	158	0	4	0	3	2	4	2	0	123	3D	0	X	1	3	X	6	28
251	20	2	1	8	5	4	0	2	2	4	2	1	2	38	0	X	1	3	X	6	29
252	20	2	4	2	9	1	5	2	2	5	2	8	23	6E	0	X	1	1	X	10	30
253	20	2	4	5	0	2	1	2	2	5	1	0	123	0	0	X	1	3	X	58	31
254	20	2	12	89	8	1	2	1	1	4	1	0	0	5E	0	X	1	3	X	43	32
255	20	2	12	89	23	1	2	1	1	4	1	8	2	5E	0	X	1	3	X	43	33
256	20	2	24	2	09D	2	0	2	1	5	0	0	23	3	0	X	1	0	X	10	34
257	20	2	1	24	8	4	1	1	0	5	0	0	0	3E	0	X	1	3	X	54	35
258	20	2	4	2	239	2	5	2	1	3	0	8	2	6F	0	X	1	3	X	10	36
259	20	2	14	24	D	0	5	1	1	4	1	0	0	367E	0	X	1	2	X	10	37
260	20	2	4	2	08B	0	5	2	1	2	0	13	23	3	0	X	1	0	X	10	38
261	20	2	0	6	1	4	1	2	1	7	2	0	0	AD	0	X	1	3	X	46	39
262	20	2	1	69	0	2	1	1	2	4	1	0	123	5	0	X	1	3	X	2	40
263	20	2	12	16A	1	3	2	2	2	4	2	0	1	5DE	2	X	1	3	X	47	41
264	20	2	1	8A	3D	4	1	2	2	4	2	8	1	8	0	X	1	3	X	42	42
265	20	2	124	057A	0	0	2	2	1	5	1	0	123	58	0	X	1	3	X	1	43
266	20	2	14	18A	9	2	2	2	2	3	1	7	1	B	0	X	1	3	X	33	44
267	20	2	124	18A	8	3	4	2	2	4	2	0	0	5	0	X	1	2	X	39	45
268	20	2	125	689	0D	2	5	2	1	4	1	0	123	35E	0	X	1	0	X	41	46
269	20	2	14	1568	0	4	0	3	3	6	2	0	123	3D	0	X	1	3	X	41	47
270	20	2	4	1A	3	4	1	3	3	3	2	0	0	7D	0	X	1	3	X	42	48
271	20	2	4	158	29	3	4	2	2	4	1	0	2	0	0	X	1	3	X	34	49
272	20	2	234	678	1B	0	4	2	2	3	2	0	0	3D	0	X	1	3	X	1	50
273	20	2	14	48A	149E	0	0	0	0	4	1	8	1	58	0	X	0	2	X	11	51
274	20	2	1	78	49C	2	2	1	1	4	1	8	2	58	0	X	1	3	X	15	52
275	20	2	1	9	1	1	2	2	2	4	1	0	3	58	0	X	1	3	X	17	53

276-305

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
276	20	2	1	9	29B	4	1	2	2	4	1	8	2	5B	0	X	1	3	X	21	54
277	20	2	14	48A	19E	4	1	2	1	4	1	7	1	58	0	X	1	3	X	24	55
278	20	2	4	14A	19C	2	1	3	3	5	2	7	1	D	0	X	1	3	X	24	56
279	20	2	1	8	148	4	1	2	2	5	1	7	0	8	0	X	1	3	X	24	57
280	20	2	1	2	68C	0	5	3	2	7	2	7	3	5D	0	X	1	3	X	10	58
281	20	2	1	2	19	0	5	3	1	6	2	8	2	5D	0	X	1	3	X	10	59
282	20	2	1	2	09C	0	5	2	0	8	0	7	3	5F	1	X	1	3	X	10	60
283	20	2	1	2	789	0	5	3	1	8	1	7	12	5D	1	X	1	3	X	10	61
284	20	2	12	2	D	0	5	3	2	6	1	0	0	5D	0	X	1	3	X	10	62
285	20	2	124	5	1	4	0	2	1	8	1	0	23	1	0	X	1	3	X	23	63
286	20	2	1	468A	9	4	1	2	2	4	2	0	1	8	0	X	1	3	X	59	64
287	20	2	1	16A	1	3	1	2	2	8	1	4	13	E	2	X	1	3	X	49	65
288	20	2	25	6A	1	2	4	2	2	4	2	2	3	5	0	X	1	3	X	51	66
289	20	2	12	69	4A	1	5	1	1	4	1	0	0	8	1	X	1	3	X	25	67
290	202	0		69	4C	4	5	2	0	8	0	0	23	5	0	X	1	3	X	25	68
291	202	0		69	0	4	0	2	0	8	0	0	123	5	0	X	1	3	X	50	69
292	202	124		368	2589	234	1	2	2	7	1	5	2	3A	0	X	1	3	X	63	70
293	202	12		8	0	3	1	2	0	4	0	0	3	5	0	X	1	3	X	48	71
294	202	1		###	9	4	1	2	2	6	1	0	2	5	0	X	1	3	X	4	72
295	202	1		4	6	3	1	2	1	7	1	0	0	E	0	X	1	3	X	4	73
296	202	0		6A	39E	4	1	1	2	7	2	8	12	2	0	X	1	3	X	30	74
297	202	1		48	C	3	1	2	1	8	1	0	1	E	0	X	1	3	X	61	75
298	202	34		67	1	1	2	1	1	5	1	0	3	1	0	X	1	3	X	53	76
299	202	24		468A	08E	2	5	1	1	4	1	2	1	6	0	X	1	0	X	52	77
300	202	14		38	08D	4	1	2	1	4	1	8	3	E	1	X	1	3	X	62	78
301	202	234		678	9	4	0	3	2	5	1	0	12	36	0	X	1	3	X	33	79
302	202	0		9	0	1	1	2	2	7	1	0	123	5DE	0	X	1	3	X	58	80
303	202	4		1678	0	2	2	2	2	2	1	0	123	5DE	0	X	1	3	X	1	81
304	202	1		2	0	1	5	3	3	7	2	0	123	3D	0	X	1	3	X	20	82
305	202	1		48A	67	4	0	2	1	6	1	8	13	ED	0	X	1	3	X	26	83

306-337

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
306	20	2	1	48A	67	1	1	2	1	7	1	8	13	ED	1	X	1	3	X	26	84
307	20	2	0	6A	3B	2	1	2	1	7	1	1	2	5DE	0	X	1	3	X	21	85
308	20	2	0	3A	9	4	0	2	2	7	2	0	1	5	0	X	1	3	X	45	86
309	20	2	0	39	3D	4	1	2	0	8	0	8	0	123	0	X	1	3	X	45	87
310	20	2	124	578	9BD	4	1	2	0	7	0	1	1	34	0	X	1	3	X	44	88
311	20	2	1	6	0	1	5	1	1	6	1	0	123	E	0	X	1	3	X	38	89
312	20	2	1	168	0	3	1	2	1	7	1	0	123	E	0	X	1	3	X	38	90
313	20	2	0	9	5	1	2	2	0	7	0	0	3	4	0	X	1	3	X	37	91
314	20	2	1	1A	2B	3	1	1	1	7	1	8	23	5D	0	X	1	3	X	30	93
315	20	2	14	18A	4	3	1	3	3	6	2	7	0	3D	0	X	1	3	X	30	94
316	20	2	0	69	5A	3	1	2	2	8	1	8	1	9	0	X	1	3	X	27	95
317	20	2	0	9	4	4	0	2	0	8	0	0	23	E	0	X	1	3	X	33	96
318	20	2	0	16	0	3	1	3	3	6	2	0	123	D	0	X	0	3	X	33	97
319	20	2	1	8	5	4	0	2	2	6	2	0	0	D	2	X	1	3	X	33	98
320	20	2	0	6	01A8	1	1	3	2	8	1	3	1	D	1	X	1	3	X	33	99
321	20	2	14	8	4	4	0	2	2	4	2	8	1	3	0	X	1	3	X	25	100
322	20	2	14	8A	4E	4	1	2	1	6	1	0	1	3	1	X	1	3	X	25	101
323	20	2	14	8A	1	4	1	2	1	6	1	8	1	5	0	X	1	3	X	25	102
324	20	2	234	68	0A	3	1	2	2	5	2	8	123	3	0	X	1	3	X	21	103
325	20	2	0	A	3	1	1	3	3	7	1	8	1	5	0	X	1	3	X	21	104
326	20	2	3	79	D	0	5	0	0	4	1	0	0	6	0	X	1	0	X	34	105
327	20	2	234	578	0	2	2	2	3	5	2	0	123	3	0	X	1	3	X	55	106
328	20	2	1	8A	8	1	1	1	1	7	1	0	1	9	0	X	0	3	X	56	107
329	20	2	1	18A	7	3	1	1	1	7	1	0	1	5	0	X	0	3	X	56	108
330	20	2	0	9	0	0	2	2	0	8	0	0	123	3	0	X	1	3	X	37	109
331	20	2	23	9	0	2	2	1	1	5	1	0	123	3	0	X	1	3	X	58	110
332	20	2	124	3	23B	2	1	2	2	4	2	0	2	D	0	X	1	3	X	1	400
333	20	2	14	168	0	3	1	2	2	4	2	0	123	E	0	X	1	3	X	1	401
334	20	2	12	469	0	1	2	2	1	45	1	0	123	5	0	X	1	3	X	2	402
335	20	2	12	69	0	12	15	1	1	4	1	0	123	5AD	0	X	1	3	X	2	403
336	20	2	124	69	0	2	4	2	1	4	0	0	123	67	0	X	0	3	X	50	404
337	20	2	0	9	0	1	2	1	1	7	1	0	123	5	0	X	1	3	X	56	405

338-370

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
338	20	2	0	57	0	2	5	2	2	7	1	0	123	3D	0	X	1	3	X	56	406
339	20	2	0D	9	0	23	2	1	1	7	1	0	123	5E	0	X	1	0	X	61	407
340	20	2	0	69	0	1	5	2	1	78	1	0	123	D	0	X	1	3	X	61	408
341	20	2	0	58	9	4	5	2	2	6	2	0	12	F	0	X	1	3	X	70	409
342	20	2	124	5	4	2	2	3	3	4	1	8	2	E	0	X	1	3	X	18	426
343	20	2	0	46A	15	1	5	2	2	8	2	8	0	A	0	X	1	3	X	73	439
344	20	2	1	146	A	3	5	12	1	3	1	7	0	8	0	X	1	3	X	67	454
345	20	2	1	16	0	3	5	1	1	3	1	0	123	E	0	X	1	3	X	67	456
346	20	2	12	69	0	1	1	3	3	4	2	0	123	3	0	X	1	3	X	12	114
347	20	2	0	248A	0E	4	1	3	2	8	1	57	2	5B	0	X	1	3	X	21	115
348	20	2	12	68	1	4	0	2	1	4	1	6	13	F	0	X	1	3	X	25	116
349	20	2	125	9	1	1	5	2	1	4	1	0	3	3A	0	X	1	3	X	2	117
350	20	2	23	16A	3AD	2	0	2	1	4	1	AC	1	AB	0	X	1	3	X	45	118
351	20	2	0	6	0	4	0	2	2	8	1	0	123	5	0	X	1	3	X	73	440
352	20	2	1	8	5	4	0	2	2	4	2	0	0	4	0	X	1	3	X	73	441
353	20	2	4	58	0	3	5	2	1	3	1	0	123	4	0	X	1	3	X	75	446
354	20	2	0	358A	02B	3	1	2	2	7	2	4	23	DE	0	X	1	3	X	49	448
355	20	2	1	3	1234E	1	5	2	1	4	1	7	23	5AE	0	X	1	3	X	79	457
356	62	6	0	A	23	4	4	0	X	4	X	8	2	E	X	X	1	3	0	64	111
357	62	2	0	A	23	3	1	2	2	7	1	8	2	E	0	X	1	3	X	64	111
358	62	6	234	05A	B	4	5	1	X	2	X	0	0	0	X	X	1	1	0	11	113
359	62	2	234	05A	3	2	5	1	1	3	1	1	0	E	0	X	1	0	X	11	113
360	62	6	12	69	0	4	4	1	X	1	X	0	123	3	X	X	1	3	0	12	114
361	62	2	12	69	0	1	1	3	3	4	2	0	123	3	0	X	1	3	X	12	114
362	62	6	0	248A	0E	4	5	0	X	8	X	57	2	5B	X	X	1	3	0	21	115
363	62	2	0	248A	0E	4	1	3	2	8	1	57	2	5B	0	X	1	3	X	21	115
364	62	6	12	68A	14	4	5	1	X	6	X	0	13	6F	X	X	1	12	0	25	116
365	62	2	12	68	1	4	0	2	1	4	1	6	13	F	0	X	1	3	X	25	116
366	62	6	2345	59	BE	24	5	1	X	4	X	0	1	0	X	X	1	0	0	2	117
367	62	2	125	9	1	1	5	2	1	4	1	0	3	3A	0	X	1	3	X	2	117
368	62	6	124	9	D	4	5	1	X	2	X	0	0	A	X	X	1	0	0	45	118
369	62	2	23	16A	3AD	2	0	2	1	4	1	AC	1	AB	0	X	1	3	X	45	118
370	62	6	0	6	0	0	5	3	X	0	X	0	123	5	X	X	0	3	0	73	440

371-402

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
371	62	2	0	6	0	4	0	2	2	8	1	0	123	5	0	X	1	3	X	73	440

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
372	62	6	1	8	5	1	5	1	X	0	X	0	0	2	X	X	0	3	0	73	441
373	62	2	1	8	5	4	0	2	2	4	2	0	0	4	0	X	1	3	X	73	441
374	62	6	4	58	0	2	5	1	X	1	X	0	123	0	X	X	1	3	0	75	446
375	62	2	4	58	0	3	5	2	1	3	1	0	123	4	0	X	1	3	X	75	446
376	62	6	0	1A	02B	3	5	1	X	0	X	4	23	0	X	X	1	0	0	49	448
377	62	2	0	1A	02B	3	1	2	2	7	2	4	23	DE	0	X	1	3	X	49	448
378	62	6	1	3	1234E	1	5	0	X	2	X	7	23	5AE	X	X	1	3	1	79	457
379	62	2	1	3	1234E	1	5	2	1	4	1	7	23	5AE	0	X	1	3	X	79	457
380	63	6	12	3	0	1	5	1	X	1	X	7	3	56	0	X	1	3	X	36	92
381	63	3	12	3	3	X	X	X	1	6	1	0	3	56	1	0	X	X	X	36	92
382	63	6	1	2	B	2	5	2	X	7	X	0	0	3	X	X	1	3	0	10	112
383	63	3	1	2	B	X	X	X	2	5	2	0	0	3	0	0	X	X	X	10	112
384	63	6	23	2	4	3	5	2	X	2	X	8	2	24	X	X	1	3	0	10	201
385	63	3	23	2	2	X	X	X	0	6	0	8	2	3	0	2	X	X	X	10	201
386	63	6	25	6	4	3	5	1	X	2	X	0	2	F	X	X	1	0	0	16	202
387	63	3	25	6	4	X	X	X	1	4	1	0	2	F	0	0	X	X	X	16	202
388	63	6	25	16	B	3	5	1	X	2	X	0	2	4	X	X	1	0	0	16	203
389	63	3	25	16	0	X	X	X	0	4	0	0	123	3	2	0	X	X	X	16	203
390	63	6	5	69	5	2	5	1	X	1	X	0	0	4	X	X	1	0	0	17	204
391	63	3	5	69	5	X	X	X	1	4	0	0	0	4	0	0	X	X	X	17	204
392	63	6	5	69	2	2	5	2	X	1	X	0	2	4	X	X	1	0	0	17	205
393	63	3	5	69	2	X	X	X	2	5	1	0	2	4	0	0	X	X	X	17	205
394	63	6	12	6A	B	2	4	1	X	2	X	0	0	3	X	X	1	2	0	19	206
395	63	3	12	6A	1	X	X	X	2	4	2	6	1	5	0	0	X	X	X	19	206
396	63	6	34	1	4	2	5	2	X	2	X	0	23	4D	X	X	1	0	0	1	207
397	63	3	34	1	4	X	X	X	2	2	2	0	23	4D	0	0	X	X	X	1	207
398	63	6	2	1	0	3	5	2	X	2	X	0	123	3	X	X	1	0	0	20	208
399	63	3	2	1	0	X	X	X	0	7	1	0	123	3	0	0	X	X	X	20	208
400	63	6	34	8	9	2	5	2	X	1	X	0	1	0	X	X	1	0	0	20	209
401	63	3.	34	8	9	X	X	X	2	3	1	0	1	0	1	0	X	X	X	20	209
402	63	6	1	A	34	3	5	2	X	1	X	6	1	2	X	X	1	1	1	25	210

403-432

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
403	63	3	1	A	4A	X	X	X	2	3	2	8	1	F	0	0	X	X	X	25	210
404	63	6	25	9	9B	2	4	1	X	2	X	1	2	4	X	X	1	0	0	2	211
405	63	3	25	9	9B	X	X	X	1	5	1	1	2	4	0	0	X	X	X	2	211
406	63	6	25	9	4	2	5	2	X	2	X	8	0	46	X	X	1	0	0	35	212
407	63	3	25	9	4	X	X	X	1	2	1	8	0	46	0	0	X	X	X	35	212
408	63	6	34	38	BE	1	4	2	X	2	X	0	0	4	X	X	1	3	0	36	213
409	63	3	34	38	D	X	X	X	3	5	2	7	1	5	0	0	X	X	X	36	213
410	63	6	25	169	9	0	5	2	X	2	X	0	2	4	X	X	1	0	0	40	214
411	63	3	25	169	9	X	X	X	2	4	1	0	2	4	1	0	X	X	X	40	214
412	63	6	25	9	D	0	5	1	X	1	X	0	1	9	X	X	1	3	0	5	215
413	63	3	25	9	D	X	X	X	1	5	1	0	1	9	0	0	X	X	X	5	215
414	63	6	34	1	E	1	5	2	X	2	X	8	1	6	X	X	1	3	0	7	216
415	63	3	34	1	9	X	X	X	1	5	1	0	1	0	0	0	X	X	X	7	216
416	63	6	12	3	79	2	5	1	X	2	X	0	1	5	X	X	1	1	0	4	413
417	63	3	12	4	E	X	X	X	0	7	0	0	0	5	0	0	X	X	X	4	413
418	63	6	12	3	0	2	5	1	X	1	X	6	123	5	X	X	1	0	0	4	414
419	63	3	12	4	0	X	X	X	0	7	1	6	123	5A	0	0	X	X	X	4	414
420	63	6	2	4A	69C	2	5	1	X	2	X	8	1	3	X	X	1	0	0	26	415
421	63	3	3	4A	69C	X	X	X	1	5	0	8	1	3	0	0	X	X	X	26	415
422	63	6	4	4	39E	3	5	2	X	2	X	2	1	5	X	X	1	0	0	52	430
423	63	3	4	4	39E	X	X	X	2	4	2	2	1	5	0	0	X	X	X	52	430
424	63	6	25	8	9	0	5	1	X	2	X	0	2	3	X	X	1	0	0	40	444
425	63	3	25	8	9	X	X	X	2	4	0	0	2	3	1	0	X	X	X	40	444
426	63	6	24	5	2	0	5	2	X	1	X	0	23	0	X	X	1	2	0	40	445
427	63	3	24	5	2	X	X	X	1	4	1	0	23	0	0	0	X	X	X	40	445
428	63	6	14	5	0	0	5	2	X	2	X	0	123	0	X	X	1	2	0	20	468
429	63	3	14	5	0	X	X	X	1	3	1	0	123	8	0	0	X	X	X	20	468
430	64	6	234	1568A	0	23	5	2	X	2	X	0	3	7	X	X	1	0	0	20	301
431	64	4	234	1568A	0	X	X	X	23	7	1	0	3	7	X	0	X	X	X	20	301
432	64	6	1245	168A	9	3	5	2	X	2	X	0	2	E65	X	X	1	0	0	23	302

433-462

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
433	64	4	1245	168A	9	X	X	X	23	7	4	0	3	E65	X	0	X	X	X	23	302
434	64	6	234	168A	25	12	5	1	X	4	X	7	2	0	X	X	1	0	0	24	303
435	64	4	234	168A	2B	X	X	X	3	7	1	0	2	367	X	0	X	X	X	24	303
436	64	6	125	68	0D	2	5	3	X	1	X	2	3	97	X	X	1	0	1	32	304
437	64	4	125	68	D	X	X	X	1	7	1	0	0	69	X	1	X	X	X	32	304
438	64	6	124	68A	0	3	5	23	X	1	X	0	3	8	X	X	1	3	0	32	305
439	64	4	124	68A	0	X	X	X	1	7	3	0	3	8	X	0	X	X	X	32	305
440	64	6	25	69	1	1	5	1	X	4	X	0	3	59	X	X	1	3	1	6	306
441	64	4	25	69	1	X	X	X	1	7	1	0	3	59	X	0	X	X	X	6	306
442	64	6	12	468A	D8C4	23	5	2	X	1	X	7	1	39	X	X	1	0	1	60	307
443	64	4	12	468A	D8C4	X	X	X	0	7	3	7	1	39	X	0	X	X	X	60	307
444	64	6	12	4A	01AD	3	5	0	X	4	X	3	1	A	X	X	1	0	0	26	416
445	64	4	12	4A	C	X	X	X	0	7	1	0	0	A	0	0	X	X	X	26	416
446	64	6	1	6	0	4	5	3	X	1	X	0	123	5	X	X	1	3	0	50	421
447	64	4	1	6	0	X	X	X	1	7	1	0	123	3	X	0	X	X	X	50	421
448	64	6	1	69	0	2	5	1	X	2	X	0	123	5	X	X	1	0	1	50	422
449	64	4	1	69	0	X	X	X	2	7	1	0	123	9	X	0	X	X	X	50	422
450	64	6	1	9	0	1	5	1	X	1	X	0	123	3	X	X	1	3	0	46	424
451	64	4	1	9	0	X	X	X	1	7	1	0	123	3	X	0	X	X	X	46	424
452	64	6	1	2	0	0	5	2	X	2	X	0	123	3	X	X	1	3	0	10	425
453	64	4	1	2	0	X	X	X	0	6	3	0	123	3	X	0	X	X	X	10	425
454	64	6	4	5	B	0	5	1	X	1	X	0	0	3	X	X	1	3	0	18	427
455	64	4	4	5	B	X	X	X	3	5	0	0	0	3	X	0	X	X	X	18	427
456	64	6	12	9	0	2	5	0	X	3	X	0	123	3	X	X	1	3	0	76	447
457	64	4	12	9	0	X	X	X	1	7	1	0	123	3	X	0	X	X	X	76	447
458	64	6	1	39	235B	2	5	2	X	0	X	0	23	5E	X	X	1	3	0	77	449
459	64	4	1	39	235	X	X	X	2	6	1	0	23	5E	X	0	X	X	X	77	449
460	64	6	0	1	0	3	5	2	X	1	X	0	123	5E	X	X	1	1	0	67	455
461	64	4	0	1	0	X	X	X	2	7	1	0	123	5E	X	0	X	X	X	67	455
462	64	6	2	6	0	2	5	2	X	1	X	0	123	D	X	X	1	1	0	39	459

464-end

DATA_START

ROW	gtyp	unit	gocc	gsty	gcol	gnck	gslv	gftt	gftb	glnt	gleg	gpat	gtne	gmat	gslt	gplt	gbck	gclr	gbrs	manu	filename
464	64	6	1	6	0	2	5	1	X	2	X	0	123	0D	X	X	1	2	0	27	460
465	64	4	1	6	0	X	X	X	1	7	4	0	123	D	X	0	X	X	X	27	460
466	64	6	1	A	D	3	4	2	X	0	X	0	123	5E	X	X	1	0	0	32	462
467	64	4	1	A	0	X	X	X	1	7	1	8	0	8	X	0	X	X	X	32	462
468	64	6	1	5	78D	0	5	2	X	2	X	7	1	E	X	X	1	3	1	81	463
469	64	4	1	5	78D	X	X	X	1	7	1	7	1	E	X	0	X	X	X	81	463
470	65	6	23	138	###	1	5	2	X	1	X	8	2	937	X	X	1	3	0	10	351
471	65	5	23	138	###	X	X	X	2	3	1	8	2	937	X	0	X	X	X	10	351
472	65	6	234	148A	B	2	4	2	X	2	X	0	0	4	X	X	X	3	0	13	352
473	65	5	234	148A	B	X	X	X	3	23	1	0	0	4	X	0	X	X	X	13	352
474	65	6	234	48	2	2	5	1	X	1	X	4	2	0	X	X	1	0	0	13	353
475	65	5	234	48	2	X	X	X	1	4	3	0	2	0	0	0	X	X	X	13	353
476	65	6	23	18A	1	2	5	1	X	2	X	0	0	59E	X	X	X	0	0	10	357
477	65	5	23	18A	1	X	X	X	1	23	1	0	0	3	0	0	X	X	X	10	357
478	65	6	24	18A	41E	3	5	1	X	2	X	7	3	E	X	X	X	0	0	24	359
479	65	5	24	18A	014E	X	X	X	2	23	1	5	3	3	0	0	X	X	X	24	359
480	65	6	234	1348A	D9	3	0	1	X	1	X	7	2	3	X	X	X	2	0	24	360
481	65	5	234	1348A	017B	X	X	X	2	23	1	5	3	37	0	0	X	X	X	24	360
482	65	6	34	016A	C	2	4	1	X	1	X	0	2	7	X	X	1	3	0	68	364
483	65	5	34	016A	C	X	X	X	2	3	1	0	2	7	0	0	X	X	X	68	364
484	65	6	24	18A	7	2	5	1	X	2	X	0	1	9E	X	X	X	0	0	69	365
485	65	5	24	18A	7	X	X	X	2	23	1	0	1	9E	0	0	X	X	X	69	365

DATA_END

9) screenshot of the advice and PRIORITIZED INVENTORY SELECTIONS (figure 36)

GUIDE 2 STYLE

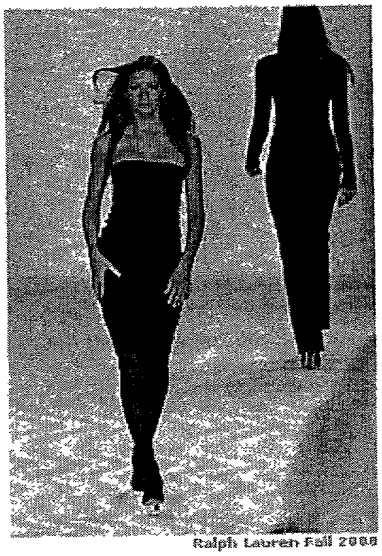
about G2S contact G2S feedback search help

what 2 wear | what 2 give | power browser | in & out | fashion forum | my G2S

your best look

Event Type: formal **Results:** 6 outfit(s) found. page #1 of 2

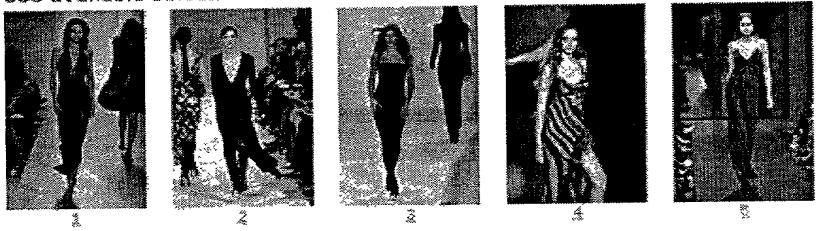
Your best look for this formal affair is a sexy evening gown.
Dress Advice: To best compliment your lean super model figure, the dress should cling closely to both your upper and lower body. Dresses bearing low cut or halter style necklines with sleeveless or strapy sleeves are stylishly appropriate for any formal affair this winter. Social etiquette for formal wear dictates long evening gowns; as such, make an entrance in a dress with an ankle length or heel length hemline. A dress that is tapered over your legs will call attention to your shapely hips and thighs. If you're feeling flirtatious, you certainly have the body to wear dresses that expose your back and bare revealing slits. The colors black, grey, blue, green, pink and gold, in light or vibrant shades (priority order) have been specifically selected to highlight your color tones, accentuate your curvaceous figure and respect this winter's formal fashion trends. While a monochromatic dress is both elegant and slenderizing, animal print and plaid/checker patterns are similarly favorable. You'll find sequins, satin and cashmere materials most becoming. Dress well.



Ralph Lauren Fall 2008

tell a friend | give us feedback

See available selections: click thumbnails below to show enlarged picture



1 2 3 4 5

Event Type: formal **Results:** 6 outfit(s) found. page #1 of 2

Select a new garment combination
Your best look is shown above. but the below garment combinations are also appropriate.

○ dresses ○ tops/skirts ○ top/pants ○ jackets/pants
○ jackets/dresses ○ jackets/skirts

print this page

what 2 wear: user input | your best look | *product detail & accessories (coming soon)*